Optimization of Video Delivery Over Opportunistic Networks

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Abhishek Kumar Thakur

ID. No. 2011PHXF0402H

Under the supervision of

Dr. Chittaranjan Hota

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
2018

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis entitled Optimization of Video Delivery Over Opportunistic Networks and submitted by Abhishek Kumar Thakur, ID No 2011PHXF0402H for award of Ph.D. of the Institute embodies the original work done by him under my supervision.

Signature of the Supervisor:

Name in capital letters: DR. CHITTARANJAN HOTA

Designation: Professor, Dept. of CSIS

Date:

Acknowledgments:

At the outset, I would like to express my deepest gratitude to Prof. Chittaranjan Hota for his guidance and support throughout the journey of my Ph.D. He mentored and motivated me to explore multiple aspects of the problem domain. Research has been a different paradigm for me when compared to other learning and professional work I had done in the past. Prof. Hota provided me with ample avenues to experiment and explore. Working with him has been a great privilege and tremendous learning experience.

My doctoral advisory committee Prof. Bhanumurti Neti and Dr. Geethakumari have always provided timely and valuable feedback. Multiple discussions with them have significantly helped my work.

I would also like to express my gratitude to other members of the faculty of CSIS with whom I have had the privilege to discuss my ideas and collect feedback. Thanks a lot to Mr. KCS Murti, Dr. Digambar Powar, Mr. Rakesh Prasanna Chennupati, Dr. Suvadip Batabyal, Dr. Sudeepta Mishra, Dr. Subhrakanta Panda, Dr. Aruna Malapati, Dr. R Gururaj and Dr. Tathagat Ray for their valuable inputs during multiple interactions.

Special thanks to research collaborators Dr. Pratik Narang, Dr. Jagan Mohan D, Mr. Ankesh Kumar, Mr. Amit Chawla, Ms. Aarthi Janakiraman, Mr. Sathiyanaranan R, Mr. Arnav Dhamija and Mr. Tejeshwar Reddy. It has been a privilege to work with them all. Discussions with Dr. Manoj Agarwal, Mr. Shekhar Chennupati, Mr. Akarshan Sajja and Mr. Samir Jain helped me in bringing the real world perspective to my research. I would also like to thank Mr. K Ramanchandran Pillai for his support in proof-reading and language related help.

This journey would not have been possible without the continued support and encouragement from my family – my wife Sarjana, my daughter Vrinda, my father Shri Kamdeo Thakur and my mother, Smt. Janaki Devi. Thanks a ton for supporting me and keeping me motivated all through.

Finally, I would like to thank all my teachers (both past and present) for making me what I am today. They have been the inspiration for this journey. Rather than mentioning their names individually, I would like to acknowledge their contribution by remembering the following Sanskrit shloka:

गुरुर्ब्रह्मा गुरुर्विष्णुर्गुरुर्देवो महेश्वरः ।
गुरु शाक्षात परं ब्रह्म तस्मै श्रीगुरवे नमः ॥१॥

Gurur-Brahma, Gurur-Vishnu-Gurur-Devo Maheshwarah
Guru Saakshaata Param Brahma, Tasmai Shri-Gurave Namah!

Abstract:

In this thesis, we study the design of a system that adaptively streams video over Opportunistic-networks (Opp-Nets). It also studies optimizations at application layer (payload reduction), network layer (fairness and policy enforcement) and link layer (energy savings in neighbor discovery).

The study covers designing a system that deploys scalable video coding (SVC) over Opp-Nets, where SVC uses a mix of lower frame rate, lower resolution, or lower quality to improve the quality of decoder. Such deployments call for impact study of end-to-end adaptation as well as improvements taking support from routing algorithms of network elements. The combined optimizations demonstrate an improvement up to 30% while end-to-end adaptive transmission by itself provides 10 – 16% improvements.

The thesis also covers application level bandwidth reduction by streaming only the foreground objects for video. We study a bi-direction approach to improve efficiency of background detection in a farm-like scenario. It demonstrates improved identification of foreground objects and payload reduction by almost 20 times

We also study the challenges at network level in deploying a central node in Opp-Nets, to enforce policies and improved fairness. Our algorithm improves fairness by 4% in the presence of congestion. Traffic routing via central node degrades performance by more than two times.

At link layer, we study discovery schemes for energy efficient neighbor discovery using mobility (speed) change as a trigger. This scheme of discovery combined with hibernate showed double battery life, as against other discovery algorithms.

To ensure efficacy of our approach, we present two deployments- first using open source DTN stack with a platform around it for operations, management and extensible apps in rural scenario, and the other using Android Nearby Connections to communicate SVC content over Opp-Nets. In our experiments, we streamed video for more than two weeks, using the mobile phones of twenty volunteers over a pedestrian distance of 1.2 km.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Glossary:

ADB	Android Debug Bridge
AP	Access Point
ARPU	Average Revenue Per User
AVC	Advanced Video Coding
BGS	Background Subtraction
BL	Base Layer
CIF	Common Intermediate Format
D2D	Device to Device
DASH	Dynamic Adaptive Streaming over HTTP
DPDU	DTN Protocol Data Unit (in SWiFiIC context)
DTN	Delay Tolerant Network(s)
EL	Enhancement Layer
EWMA	Exponential Weighted Moving Average
EUA	End User Application
FCC	Federal Communications Commission
FTTC	Fiber To The Curb
GPS	Global Positioning System
HEVC	High-Efficiency Video Coding
HTTP	Hyper Text Transfer Protocol
IAMAI	Internet and Mobile Association of India
IMN	Independent Mobile Nodes
IP	Internet Protocol
ITU-T	International Telecommunication Union - Telecommunication standardization sector
JSON	JavaScript Object Notation
JSVM	Joint Scalable Video Model
MANET	Mobile Ad-hoc Network(s)
MDC	Multiple Description Coding
MOG	Mixture of Gaussian
MOS	Mean Opinion Score

MPEG	Moving Picture Experts Group
MPLS	Multi-Protocol Label Switching
NAL	Network Adaptation Layer
NALU	Network Adaptation Layer Unit
NOFN	National Optical Fiber Network
Opp-Nets	Opportunistic Networks
ONE	Opportunistic Network Environment
P2P	Peer-to-Peer
PROPHET	Probabilistic routing based on prior history of encounters and transitivity
PSNR	Peak Signal to Noise Ratio
QoS	Quality of Service
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RSVP	Resource Reservation Protocol
RWP	Random-Way-Point
SDN	Software Defined Networks
SFT	San-Francisco taxi Traces
SHM	SHVC test Model
SHVC	Scalable HEVC
SNW	Spray and Wait
SOApp	SWiFiIC Operator App
SORT	SWN with Operating-point, Replica and TTL control
SPOF	Single-point-of-Failure
SUTA	SWiFiIC User terminal App
SVC	Scalable Video Coding
SWiFiIC	Sustainable Wi-Fi in Indian Context
TCP	Transmission Control Protocol
TTL	Rime-to-live
UDP	User Datagram Protocol
UGMC	User Generated Multimedia Content
USB	Universal Serial Bus
UT	User Terminal

VCL	Video Coding Layer
WDM	Working day model
XML	Extensible Markup Language

1. INTRODUCTION

These days, user-generated multimedia content (UGMC) is used, primarily for entertainment; however, the presence of affordable mobile devices has increased the volume of UMGC. For infrastructure in urban environs, UMGC is used for knowledge sharing, security, news coverage, etc. Unfortunately, in rural areas of developing countries, it leaves some gaps, reflected as lack of affordable network connectivity and high bandwidth requirements associated typically with most video content.

Delay and disruption tolerant networks (DTN) utilize node mobility to complete the network over time. The mobile nodes carry the content till it reaches the destination. Opportunistic networks (Opp-Nets) are a subset of DTN where the node mobility pattern is unpredictable. By creating multiple replicas of the content, Opp-Nets attempt to deliver the content to the destination device(s). This work focuses on optimizing video delivery over Opp-Nets, specifically studied in the context of rural farms, the findings of which can apply to other deployments short of infrastructure, e.g., wildlife monitoring, disaster response, defense, etc.

We observed that majority of video communication protocols and applications developed for urban needs do not help to enhance the productivity of rural masses. These applications meet the entertainment and communication needs, but do not provide avenues for helping them in their day-to-day farming related activities.

1.1. Motivation

1.1.1. *Network gap between rural and urban setup*

With a significant focus on deploying content distribution networks, the content transfer has significantly improved from central / cloud-based servers to users in urban areas. Similarly, for enterprise networks, with good support from network elements on features like MPLS, SDN, resilient links, etc. user-generated content has also been shared over the Internet. The enterprise and urban networks have significant revenue from end users to justify the deployment of optimum solutions.

When we move beyond the urban centers, we have a significant geographical area where infrastructure lacks, and even when service is available, it’s not affordable for carrying multimedia content. Reports in developed countries show a significant divide between urban and rural connectivity. In fact, FCC's broadband progress report for 2016 [1] for the United States of America, showed almost ten times difference between urban and rural broadband penetration (4% for rural areas vs. 39% for urban areas). Similar results are documented for other developed countries [2]. Closer home, IAMAI report for Internet penetration in India for December 2017[3] shows more than three times lower penetration for Internet mobile connectivity in rural India. Anecdotal evidence suggests that rural mobile Internet connectivity is not suitable for broadband usage in many areas. The challenges in rural environments for Internet services are primarily because of the low density of users and difficult terrain to lay backhaul links. Many countries put a cess on urban users to subsidize the deployments of rural areas. People in rural area lack technical expertise, which implies that the service delivery is sub-optimal as technicians have to travel long distances to fix the outages.

Often to avoid high cost of laying wired infrastructure in rural areas, multi-hop wireless networks are deployed. Because of the nature of spectrum and power limitations, such networks are only capable of low to moderate bandwidth and require significant expertise to administer and operate. Ad-hoc wireless networks can fulfill some of the communication needs, for the rural scenario. Ad-hoc networks use live communication between the source and the destination via multiple intermediate hops. The intermediate nodes relay the payload in close to real time. If the node density is low or nodes move frequently, ad-hoc networks will not be able to maintain end-to-end connectivity since the network gets partitioned. Over the last few years, delay tolerant networks (DTN) [4] have been used to overcome the partitioning and sparse node density related issues of ad-hoc networks. Rather than relaying the payload in real-time, DTN nodes use the concept of store-carry-forward. Thus, as the node moves across to different areas, it helps complete the network over time. This connectivity comes at the cost of increased delay.

As discussed above, Opp-Nets are a subclass of DTN where there is no explicit pattern for contact between nodes. In such cases, to improve delivery at the destination and to lower the delay, nodes may allow multiple replicas of the payload to exist on the network. Higher replication increases the processing, storage and transmission costs.

1.1.2. *Lack of localized frameworks*

Today, majority of existing applications over the Internet rely on client-server architecture, where servers are predominantly in data centers, connected through high capacity links to the Internet. Even when two devices are in the vicinity of each other, frequently they communicate via the data center which may be thousands of miles away. But, if we survey the needs of rural population, most activities are local. Unluckily, lack of local infrastructure implies that users tend to use the applications developed to meet the urban needs, or do not use any information technology-based applications.

Of late we have seen the deployment of proximity based solutions for localized file sharing and messaging (like open garden, file share apps, etc.). These solutions used the proximity of cell phones carried by users to complete the network communication over time. The delays may be of the order of hours in some cases if users are not present in the same vicinity.

Numerous daily activities in rural environment do not require real-time communication between systems. There are scenarios where communication takes place in a delay tolerant manner, especially if the current solution itself was not reacting to events in real time. E.g., in an orchard, intrusion of animals and theft of flowers or fruits can occur. Based on the value of produce, the farmer may deploy a guard. Even in such cases, the guard will not be able to monitor all areas of the farm. It is important to reduce the likelihood of theft and intrusion. By capturing the videos and relaying it opportunistically to the farm owner, affordable camera-based devices, (powered using solar panels or similar means), can act as a good deterrent.

1.1.3. *Characteristics of Video communication*

Most of the digital communication cannot tolerate data loss or data-corruption. Auditory and visual multimedia communication is an exception to this. Multimedia communication can tolerate data-loss and data-corruption while providing acceptable perceptual quality. In prior applications, video communication involved transmission between source and destination to sense the quality of network and adjust the transmission accordingly. By observing the loss-and-error rates, the source may create additional copies or use other schemes to recover from errors within the network. Similarly, based on the observed bandwidth, the source and destination(s) may communicate content with different quality levels. E.g., when the network conditions are perfect, video may be transmitted at highest possible resolution, high frame rate and with the best fidelity. On the other hand, when the bandwidth is lower, the source can generate video at a lower resolution, lower frame rate or lower fidelity.

Some of the existing solutions use scalable video coding (SVC) to transmit multiple streams from source and let the destination decide the number of streams it received. If the destination can receive all the streams, it decodes video at the best quality. When the destination cannot receive some of the streams (possibly because of bandwidth limitations), the destination tunes only for lower SVC streams and decodes the video at lower quality.

End-to-end feedback for communication over delay and disruption tolerant networks can have a significant delay and also encounter a loss. Hence, there is a need for a more robust and flexible streaming solution for such networks.

1.1.4. *Emerging trends in opportunistic video capture and communication*

Recently we see an increase in opportunistic capture of content in multiple domains like law enforcement, disaster response, transport, defense, wildlife, agriculture, etc. Here the communication delay may be of the order of minutes to hours. Affordable smartphones and other portable devices with integrated camera have helped increase this trend. [5,6] have explored multimedia applications, which are both delay and loss tolerant for disaster management scenarios. The video and images from body-mounted cameras of members of disaster response team are shared back opportunistically. In some of the cases an advance team surveys the disaster zone and relays back the multimedia content for other team members to respond appropriately.

Additional scenarios are becoming feasible thanks to affordable smartphones with integrated camera, where opportunistic capture and communication of content can be used for better user engagement or compliance with law and order. In India, we frequently have large transitory gatherings for scenarios like religious events (Kumbh mela) or demonstrations and procession. In the first case (religious events) distribution of information will help participants to plan their activities in a better manner. In case of demonstration and processions, such capture and communication scheme can be a very affordable deterrent for participants to abide by the rules and policies.

1.2. Sample scenario for communicating multimedia content

In this thesis, we study optimization for transfer of video feeds from cameras monitoring a farm. The cameras are portable devices with Wi-Fi / Bluetooth or similar wireless connectivity. They are powered by solar panel and are possibly mounted on poles to get a view of the larger expanse of farms. The video feed is used for posterior viewing and action, rather than real-time processing. E.g., the farmers may periodically look at the video feed (say once a day), and if they find any issue with the current state, they look at the video collected over prior intervals. This ensures that the trespassers are discouraged from damaging the farm, or if the damage has happened, the erring party is held accountable, based on the video captured by the system.

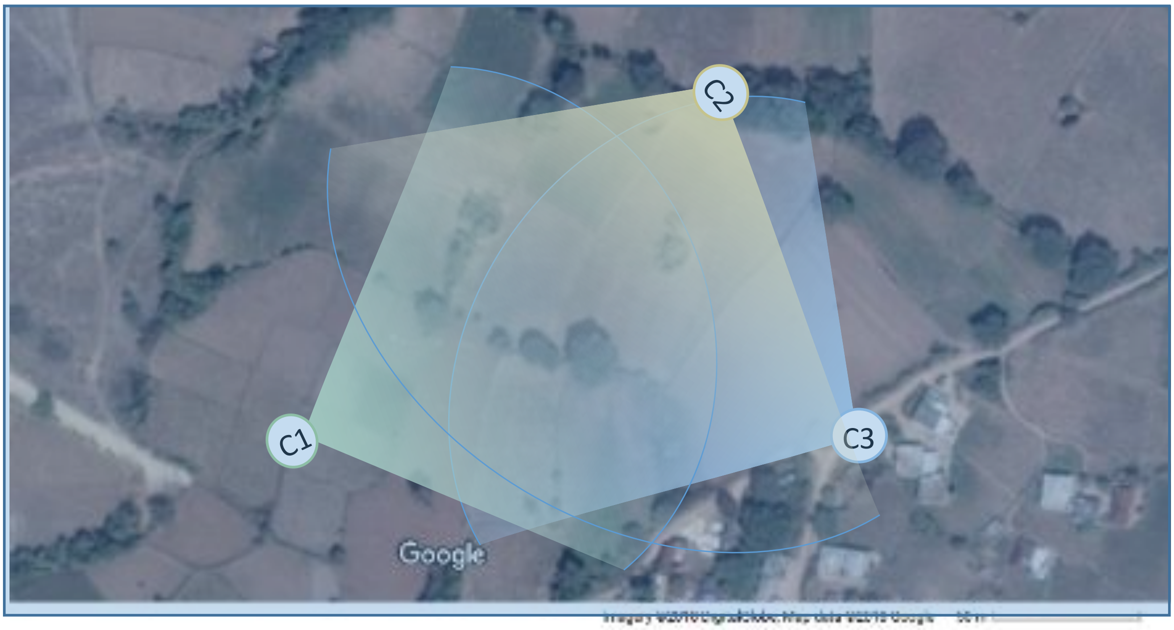


Figure 1.1 Field of view of three cameras monitoring farmland with mud path etc.

Figure 1.1 shows a possible three camera setup for capturing videos over an expanse of farmlands in Guntipally / Govindpur village near Wargal in Telangana state (background Imagery ©2018 DigitalGlobe, Map data ©2018 Google). The camera system can be powered using battery charged by solar panels. People traveling (pedestrian or in cycles) across the farms help complete the network in a delay tolerant manner.

We can apply similar scenario to other domains. The specific farm scenario provides following three key characteristics of interest to this thesis:

A. It streams data over a long interval (weeks to months).
B. Rural environment has sparse node density, especially in the vicinity of capture device.
C. Video content tends to have very little change over time, though there are challenges like the movement of branches etc.

1.3. Research problems

For the sample scenario mentioned in the preceding section, the video will be communicated by Opp-Nets using the concept of store-carry-forward. In the above context, following is the broad research problem at the core of our thesis:

"How to encode and transmit video captured in farms over Opportunistic Networks?"

Encoding and transmission of the content itself can involve multiple approaches. For example, compressed content can be packaged frame by frame or in segments of a few frames each. Since our scenario is delay tolerant, chunk-based compression provides lower overheads and hence higher efficiency. When we transmit the whole chunk as a single monolithic unit, in the presence of congestion the whole unit may be lost. To avoid congestion loss, the source can reduce bandwidth demand by utilizing more aggressive compression leading to degradation of video quality at the destination. This implies we need to devise a mechanism that adapts the video transmission quality to the network capabilities. Hence the first research problem is:

a. How to encode and transmit video on Opp-Nets such that we make efficient use of network resources?

Since the farm video will have very little change over time, it will be possible to discard parts of the video that have not changed in information. For example, if for a few minutes there are no changes, the next transmission can be skipped, or a small payload can be generated to communicate the absence of change. Similarly, if half of the video frame had a flat field and did not have any motion associated with it, it could be skipped from the transmission for the next chunk. Our next research problem is:

b. How can we discard some parts of the video, without losing perceptual information?

In rural setups, farms may be away from human habitation. Moreover, the spread of houses is not uniform. There may be a few small hamlets, while some areas may be densely populated. This can cause a significant drop in delivery rate for feeds from some of the camera systems, or for feeds to a farmer staying in a remote hamlet. The system needs to provide fairness for such users. The third research problem is:

c. How can we measure and improve the fairness of service across the nodes?

While the video source and destination nodes spend energy in video capture, encoding/decoding, and transmission, other nodes primarily spend energy on communication. In rural environments, since node density can vary, significant energy will be spent on discovering other nodes. This brings us to our final research problem:

d. How can we minimize the energy spent on node discovery across different node density, while ensuring that we do not impact the routing of payload?

1.4. Objectives of the thesis

Based on the research problems discussed in the last section, the major objectives of the research are as follows:

Obj. 1. Design and implementation of a video compression and transmission scheme to use Opp-Net resources efficiently.

Obj. 2. Devise a scheme to discard parts of the video, which has perceptually static content.

Obj. 3. Design and implement a routing scheme that improves fairness in delivery for nodes in the network.

Obj. 4. Devise a scheme to efficiently discover contacts while minimizing energy usage.

Table 1.1 DTN stack and its application to various objectives of our thesis

DTN Stack Layer	Objective 1	Objective 2	Objective 3	Objective 4
Application	Implementation	Primary implementation		
Bundle	Payload mapping			
Transport	Not-Applicable			
Network	Routing impact		Primary implementation	
Data Link				Primary implementation
Physical	Not in scope for our work			

1.5. Scope and assumptions

While we have used farm monitoring as the guiding example for this thesis, one can apply the design elements to other scenarios too. For large transitionary gatherings, like Kumbh-Mela in India, call data and text messages have been used to study the population dynamics [7]. In such scenarios, the opportunistically collected video provides excellent input for trend analysis, including posterior monitoring and investigations [5,8]. Monitoring of events in remote locations (e.g., elections or exams in sparsely populated areas) is another application where such an approach, backed by tamper-resistant local storage for a few hours can provide an effective enhancement to ensure that no malpractices take place. We can also use such opportunistically recorded content, for deep learning and analytics to improve forecast and planning for future. Besides multiple applications in defense, responding to disaster recovery scenarios, etc. will have a similar set of challenges, though the video capture and transmission may be for shorter intervals.

This thesis is an overlap of two domains – Opportunistic Networks and Multimedia Systems. Both are significantly wide domains on their own. Following is a subset of the constraints assumed in this thesis to keep the scope limited:

1. All participating entities are assumed to be altruistic, and hence security is not considered a concern for this thesis.

2. We have not considered audio compression.
3. Video feeds are assumed to be coming from fixed camera views. Mobile cameras (mounted on human body, vehicles, drones, etc.) or fixed camera systems with pan-tilt-zoom, etc. cannot utilize parts of the thesis that assume fixed camera view.
4. We assume that there is significant trust in the operating environment that the devices are not stolen or damaged. While a system can be designed to generate alerts for theft to portable devices, we do not explore it in this thesis.
5. The video feed can be relayed outside the village using gateway nodes to regular Internet-based communication. We do not explore such extensions in this thesis.
6. In keeping with the idea that Opp-Nets will not have predetermined contact patterns, we have skipped single copy routing involving custody. Similarly, we have not attempted to utilize DTN routing approaches that explore relationships across nodes.

1.6. Structure of the thesis

Next chapter covers the background work for video transmission over opportunistic networks. The Third chapter covers the tools used for experiments and the key metrics used for performance evaluation. The Fourth chapter discusses SORT as a solution to first Obj. 1. SORT uses scalable video coding for transmission over opportunistic networks. In this chapter, we also study the benefits from adaptation at end-hosts, as well as from optimization with the network. Chapter Five covers Obj. 2 and uses a bi-directional approach for improving the accuracy of foreground detection using background subtraction scheme. Chapter Six covers Obj.3. Using a central node, it studies improvement in fairness. Chapter Seven covers energy optimization during node discovery to address Obj. 4. Chapter Eight covers detail of two deployments that demonstrate sharing of multimedia content. Chapter Nine concludes by providing a recap of the thesis and discussing possible future work.

2. Background and Related Work

In this chapter, we cover the background concerning evolution in networking technologies as well as of multimedia communication. We study scalable video coding and how video can be communicated over Opp-Nets.

2.1. Evolution of Networks

Early Internet research and development was predominantly in academic realm involving static hosts. The Arpanet and subsequently early Internet provided connectivity for hosts within academic institutes using simple network components from service providers. Some of the networks (e.g., Aloha) used wireless technologies, but with static nodes. Satellite-based communication was also used to connect remote locations with slightly longer delays.

Later on, mobile telecommunication networks allowed use of portable devices for communication, starting from car-phones and gradually moving to portable smartphones. In the present era, when we talk of Internet-of-things and 5G technologies, the end devices are getting more and more portable and pervasive.

2.2. Categorization of Networks

While there are multiple ways of classifying networks, Fig. 2.1 captures a simplified classification of networks based on the presence of infrastructure elements and mobility aspects of entities within the networks. Real networks may be hybrid. E.g., Bluetooth audio headset for cellular phone call connected via telecom base station will have “fixed network but mobile end host” till the mobile phone, while it would be “mobile network and mobile host” between mobile phone and the Bluetooth headset.

Note that this thesis applies to category 7 in Figure 2.1. All the same, let us discuss all categories briefly to understand the challenges for category 7. Category 1 covers the networks as they existed in early Arpanet or with wired telephone lines. A service provider typically manages the infrastructure with significant technical expertise. It utilized fixed end hosts with permanent connectivity or predictable connectivity (e.g., dial-up modems). Furthermore, network components like telephone exchanges and links connecting them are static. Category two allows mobility on the hop connecting end-host to the network.

Base-stations for telecom service providers and access points (AP) for WiFi, allowing portable devices to connect to wired backbone, are typical examples of category two. The end hosts may have limited mobility in case of simple access points or may travel across the country using different base stations of the service provider(s). Category three covers networks which rely on parts of infrastructure that are mobile. Google's Loon project, satellite-based communication, wireless access points within a flying airplane are examples of such network. Note that, in case of access point providing Wi-Fi connectivity within the airplane, both the AP and end-user devices are mobile.

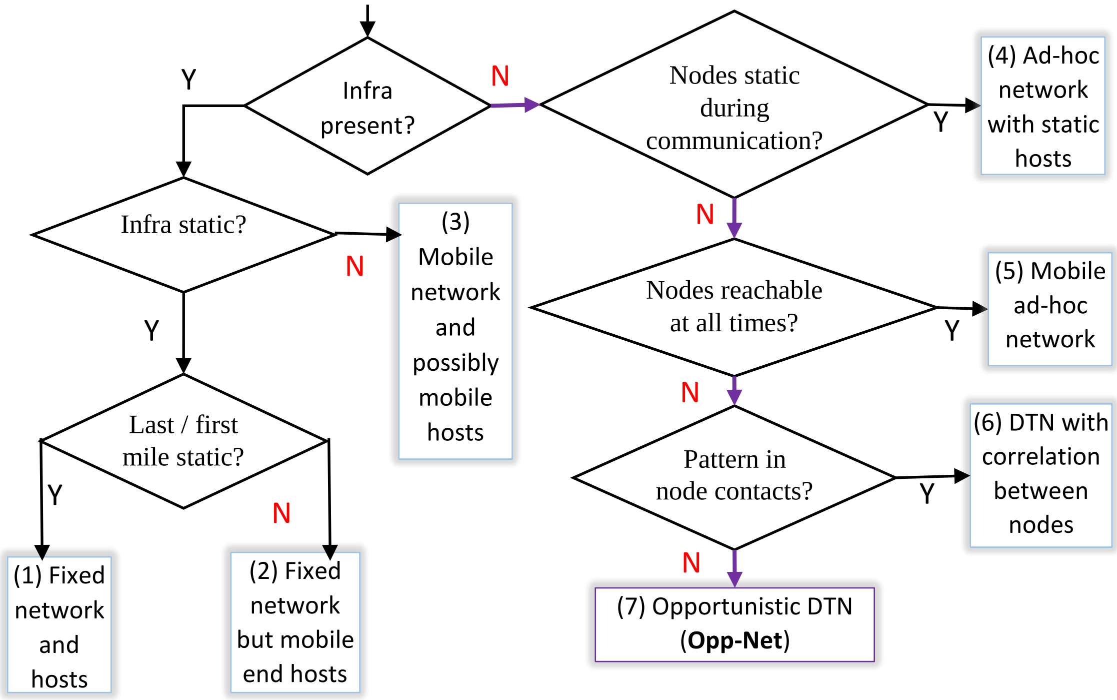

Infra present?
Y
N
Nodes static during communication?
Y
(4) Ad-hoc network with static hosts
Infra static?
N
(3) Mobile network and possibly mobile hosts
Y
Last / first mile static?
Y
N
(1) Fixed network and hosts
(2) Fixed network but mobile end hosts
N
Nodes reachable at all times?
Y
(5) Mobile ad-hoc network
N
Pattern in node contacts?
Y
(6) DTN with correlation between nodes
N
(7) Opportunistic DTN (**Opp-Net**)

Figure 2.1 Classification of networks on connectivity and mobility

In the absence of infrastructure, operating a network with good quality is a challenge. Ad-hoc networks with static hosts (category 4), use intermediate nodes to communicate between source and destination. Since the nodes are static, the communication is simpler as packets are routed through the same set of nodes. As nodes become mobile (category 5), the routes may keep changing. Path changes add to the complexity of communication overheads, but end-to-end paths still exist in these deployments.

When the nodes are spread over large areas or travel longer distances, they may get disconnected from other nodes. In such scenarios, as nodes have contacts with one another they communicate and help complete the network over time. If we know the pattern of contacts among the nodes in advance (e.g., satellites in inter-planetary networks, or trains following a fixed schedule), the communication and routing can be planned in a manner that makes optimal use of future contacts. This is category six, which expects some correlation between contacts. On the other hand, if there are no explicit patterns or the correlation is weak, we cannot plan for contacts in advance. In such cases, the communication is done opportunistically.

Note that except for category 6 and 7, all other networks have an end-to-end delay which is of the order of seconds or lower. We delve deeper into category 6 and 7 when we discuss DTN in a subsequent section.

2.3. Evolution of video communication

Similar to the evolution of networks, video communication technology has also progressed. Very early video communication systems used analog signals for television broadcasts. The systems were upgraded to digital video networks, which typically broadcast videos for prior published content and one-way broadcast of live events. Digital TV offerings (including digital cable TV, Satellite-based TV services) are early examples of this. Production houses with significant expertise were the source of these video feeds. Subsequently, as Internet penetration increased, other platforms like Netflix, YouTube, etc. started providing content individually to each user (i.e., unicast). Parallel to this, real-time video communication also evolved, initially in one-to-one mode and later in conference mode, supporting multiple sources. Examples of such communication platforms are WebEx, Skype, FaceTime, WhatsApp, Duo, etc. In all these scenarios, end users view the video on heterogeneous platforms connected to, over a variety of networks. Since the display resolution, buffering and decoding capabilities may vary, we have to adapt the original video for different receivers.

In the last few years, emphasis on user-generated content has increased. The user-generated content is captured and edited, with significant attention, and the relevant parts are shared with others using the regular Internet. If the network connectivity and resources for video capture are available, we also have numerous platforms for live streaming (like Periscope, Facebook or YouTube). They allow users to broadcast their content close to real-time. There is tremendous demand for such authentic and unedited content [9]. These platforms need significant resources like high network bandwidth, excellent processing power, and stable network connectivity.

A new category of user-generated multimedia content, where capture and transmission are done on a best-effort basis, has evolved over the last few years. It does not have a strict expectation of bandwidth and quality support from the network. Such systems are intended for providing information in challenging environments like disaster response, policing, etc. Often the video captured may be analyzed posterior, to react/respond to such events in a better manner over time.

2.4. Classification for video communication

Table 2.1 captures multiple categories for video communication. Our thesis focuses on unicast of content from multiple capture devices in a pseudo-live manner. Open source solution, using single video stream from each source, over the disconnected network is studied. The communication happens through overlay, which is on top of DTN like networks involving wither Wi-Fi direct or Bluetooth.

Table 2.1 Alternative classifications for digital video communication

Classification Theme	Possible categories
Video Source	◦ User-generated ◦ Production house
Delay from capture to playback	◦ Live ◦ Pseudo-live ◦ Delayed
Link layer technologies	◦ Licensed: DVB-T, DVB-S, Telecom networks (3G, 4G, 5G) ◦ Unlicensed: Wi-Fi with the access point; **Wi-Fi – Ad-hoc or Wi-Fi Direct; Bluetooth**
Transport	◦ UDP / IP (with RTP / RTCP etc.) ◦ TCP / IP (DASH etc.) ◦ Peer-to-peer over IP or similar **overlay at a different layer**
Expectation from Network	◦ The strict expectation of bandwidth and delay (MPLS etc.)

	◦ The expectation of a bound for lowest bandwidth and max delay. ◦ No bounds for bandwidth and delay
Number of streams from source	◦ Single video stream ◦ Stereo – two video streams ◦ Multiple – more than two streams including spherical view etc.
Source to receiver mapping	◦ **1-to-1**, 1-to-N, **N-to-1**, N-to-M
Mobility	◦ Both ends static ◦ Source static, destination mobile ◦ Source mobile, destination static ◦ Both ends mobile
Streaming platform	◦ Open source tools to capture and stream ◦ Proprietary solutions
Quality Adaptation	◦ Non-Adaptive single stream from the source ◦ Adaptation within the network (transcoding) ◦ Single adaptive stream from the source ◦ Multiple independent quality streams from source ◦ Multiple hierarchically related quality streams from source
Mapping of video to Payload	◦ One video frame to multiple data units ◦ One frame to one data unit ◦ Multiple frames (a chunk of video) to one data unit.
Digital rights and encryption	◦ Copyrighted – open access ◦ Copyrighted – encrypted ◦ Free to access and redistribute

Besides the above classification, as we study the objectives in further details, we also encounter some solution-specific options. For example, choice of compression scheme, alternatives for foreground detection, etc. are specific topics limited to different objectives; hence they are not included in this section.

2.5. Routing in DTN

Nodes in DTN communicate with each other using intermediate nodes. If the sole purpose of intermediate nodes is to help in communication, they are called auxiliary nodes. On the other hand, we may also have DTN where there are no auxiliary nodes, in which case all nodes may generate and receive payload. Such DTN routing is commonly referred to as Independent Mobile Nodes (IMN) based routing. In our thesis, we primarily consider Opp-Nets with routing involving IMN. In some specific chapters, we also consider infrastructure support involving a central node which is generally static.

Once the application layer provides a payload (bundles in DTN parlance) for transmission over DTN, the node waits for contacts with other nodes. The discovery of contact is a challenge in itself since the nodes will waste energy if they check on a continuous basis. In 6 we study in detail the challenges with contact discovery. After contact discovery, DTN routing determines which bundles to transmit and in what order. Since the contact duration is not fixed, some of the transfers may be aborted mid-way.

Figure 2.2 is a simplified DTN routing engine based on ref [10].

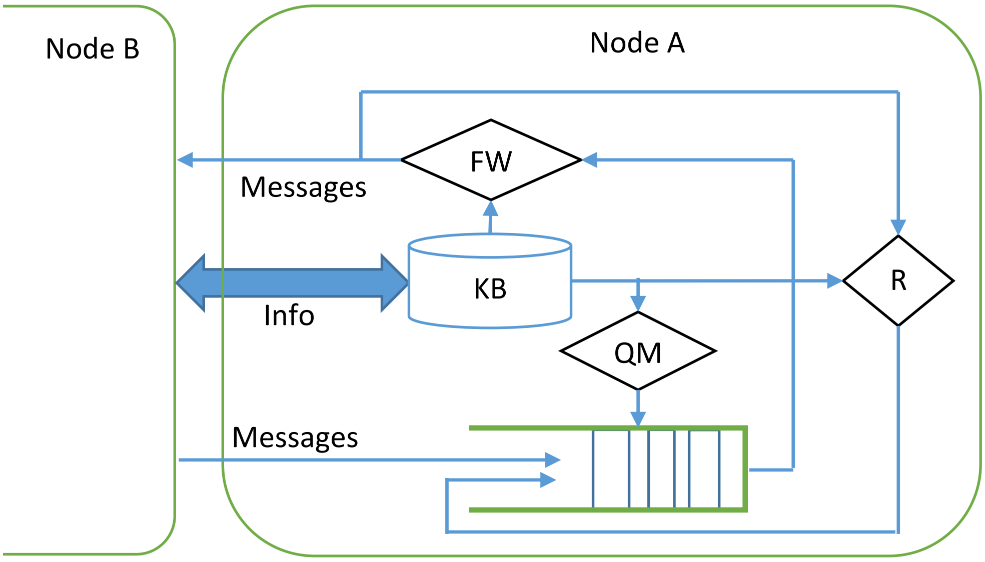


Figure 2.2 Simplified DTN routing blocks

Immediately after discovery, the nodes share their knowledge base (KB) including a list of delivered messages, the messages they have, etc. Updates in KB may trigger queue management (QM) module to modify messages in the queue such that the order of forwarding is optimized, and delivered messages are deleted. Subsequently, bundles from the queue are forwarded (FW) to the other node, and replication (R) meta-data about the bundles is updated. Note that this simplified representation only assumes that two nodes are in contact with each other. There are chances that a node is in contact with multiple other nodes. For the sake of simplicity, we are not considering such situations in this discussion.

Cao et al. in [11] provide an excellent overview of numerous DTN routing protocols. One of the classifications involves single-copy vs. multiple-copy routing. Single copy routing frequently involves custody transfer of bundle and works for a controlled environment with oracle based (or strongly predictable) connectivity – e.g., deep space networks or public transport systems that follow a strict schedule. Such DTN deployments can use single-copy based routing, as one can reasonably predict the future location, of the satellites or vehicles with high confidence. At the other extreme of single-copy, routing is epidemic routing, wherein the replica of original bundles is shared (relayed), at each contact on the assumption that one of the nodes will eventually get in contact with the destination and deliver the data. This approach works well for opportunistic networks where little correlation exists between node contacts.

One should note that epidemic routing works well when payloads are small, or the count of nodes is less. As the payload on networks increases or node density increases, epidemic routing creates significant overhead on relaying bundles (higher node count) and usage of buffers to store the bundles (higher payload). In case some correlation exists in the contact patterns of nodes, many proposals exist to minimize the number of replicas. E.g., multiple social contacts based routing approaches attempt to exploit features like community, between-ness, centrality, etc. when deciding whether to relay a bundle or not. Some other approaches rely on the history of encounter times and transitivity of the same between nodes (e.g., Prophet). Another group of algorithms attempts to solve the routing challenge as a resource optimization problem across nodes in a distributed manner (e.g., RAPID[12], MaxProp[13], etc.). These approaches try to find the utility of the other node in successful delivery of the content and give good results when such a correlation exists. Computing and sharing of information of prior contacts add to the complexity of these protocols. Furthermore, these utility-based approaches expect the prior correlation to hold for future.

Another aspect of queue-management in simplified routing block relates to buffer maintenance. It can be proactive or reactive. In case of proactive maintenance, queues are not allowed to take up the complete buffer space by proactively deleting messages. In case of reactive maintenance, when the buffer gets full (or almost full), space is created by deleting old messages, based on the size of new messages. Furthermore, in the absence of buffer-space, the queue management module can either delete older messages based on some criteria or stop accepting new messages. The choice of queue management policy also affects the performance of DTN. For our thesis, we have used the simple approach of reactive queue management, wherein prior messages are deleted to create space for new incoming messages.

One of the most widely used approaches to limit the overheads of Epidemic routing is using spray-and-wait (SNW) [14]. Here the source application decides the maximum number of copies that can exist on the network (denoted as L). In essence, it leaves control to the end-application similar to the end-to-end argument [15]. SNW allows any node to relay content only if its replica count is more than one. When replica count is one, a node can only deliver the content to the destination. When compared to epidemic routing, there is only a small overhead of tracking the notional value of L for each bundle. While sharing content to a third node, SNW can use two approaches, that is, binary-SNW or linear-SNW. Binary version gives half the copy count to the other node on contact, while linear-SNW gives only one copy to the other node. Linear-SNW works well only if the source node has a significantly high number of contacts. Our thesis predominantly uses binary-SNW.

Sobin et al. in [16] provide an overview of DTN routing for information-centric networks (ICN). In addition to regular DTN routing, they have also explored aspects of publisher-subscriber model and multicast in ICN. Note that our work is not targeting to solve problems of multicast, and is using simple client-server (or source-destination) model for transmitting videos from one source to one destination.

2.6. Video compression and encoding

The raw video takes tremendous bandwidth and hence needs to be compressed or encoded. The choice of encoder provides a trade-off between three things - size of the compressed content, computation and storage complexity during encoding/decoding and distortion in the video post-decoding. As computation and storage capabilities have increased, the newer encoding schemes use algorithms, which have a higher demand for computation or storage. Video/visual coding experts group of ITU-T developed the early digital video compression standards and published them as H-26X series with the primary aim of real-time video communication. Moving picture experts group (MPEG) developed compression schemes for pre-recorded content. Subsequently, the two teams have together combined and developed some of the encoding standards. Presently, high-efficiency video coding (HEVC or H-265) is among the state of the art video compression standards published by this combined team [17]. Earlier encoding standard is called advanced video coding (AVC/H264)[18].

For heterogeneous devices and networks, video compressed at one resolution or bit-rate may not give optimal quality for a variety of receivers. To solve this problem, three broad approaches exist – 1) *transcoding* on the network, 2) *simulcast* and 3) *SVC* and multiple-description-coding (*MDC*). For *transcoding*, the network needs significant computing resources to adapt contents to the receivers needs. *Simulcast* involves simultaneous transmission from the source on to multiple channels (e.g., using IP multicast, link layer or application layer multicast, etc.). The simulcast receiver selects optimal channel based on its capabilities and the current network performance. For pre-recorded content, there is an alternative to simulcast where the receiver pulls the content. The source publishes content in chunks, at different bit rates. The receiver first downloads details about the chunks. It then downloads the first chunk and based on the time taken to download the prior chunk, adapts the bit-rates for subsequent chunks [19]. *SVC* generates multiple layers of video. The base layer carries data for the lowest resolution (spatial), lower frame-rate (temporal) and highest quantization scale (lower quality). Subsequent enhancement layers help improve the spatial, temporal or quality aspects. Quality improvement is also called an improvement in signal-to-noise ratio (SNR). An alternative to SVC is *MDC*, which has been analyzed in detail by Kazemi et al. in [20]. MDC includes redundant information in each stream such that receiver can decode any of the streams independently. Destination node can further improve the decoded video quality, as and when it receives more MDC streams.

For opportunistic networks, we cannot rely on network infrastructure for transcoding. The transcoder may become a single point of failure. This scheme expects relay of content via transcoding node, thus increasing the delivery path length. The delivery quality (delay and loss) will degrade because of longer path length. Given the fact that many of these network elements may be resource-constrained, computation complexity of transcoding will reduce the active lifetime of such nodes. Moreover, bandwidth from the source to the transcoder will be very high. Because of such issues, we cannot use on-network transcoding approaches for opportunistic networks. Simulcast creates additional packets for the network and may make sense in information-centric deployment for data dissemination in DTNs [16]. Even in such scenarios, the redundant information between the Simulcast sessions will be a huge overhead for the network.

As mentioned before, SVC cannot successfully decode the received higher layers, when the base layer or lower layers in SVC are not available. MDC does not suffer from such issues. Irrespective of the stream received, MDC can decode the video, because of redundant information within each stream. When comparing SVC to MDC for peer-to-peer streaming, Abboud in context of P2P [21] infers that SVC is better in scenarios where network can prioritize the lower layers. Usage of MDC (with redundant information across the stream) will cause additional overhead during replication based routing. Since our work uses higher replication to protect lower layers (BL may have double the copy counts as compared to enhancement layer 1), SVC is more suitable than MDC.

Scalable HEVC (SHVC) [22], as compared to the prior version (SVC [23]), adds enhancements to bit-depth, color gamut, and provides support for hybrid codec. Moreover, HEVC has inbuilt support for SHVC layers using high-level syntax (HLS). HLS reduces overheads of packaging the layers in SHVC when compared to SVC. In our work, we have experimented with SVC and H264 to align them with some of the reference works. Moreover, because of lower overhead of SHVC, the results for SHVC should be an improvement over SVC.

In SVC, video scaling can improve frame-rate (temporal), resolution (spatial) or quality (SNR). For temporal scaling, enhancement layers increase the frame rate. Enhancement layers for spatial scaling increase resolution for the frames. SNR scaling reduces the quantization scale for enhancement layers, thus reducing the residual error for decoder. We can combine one or more of these scaling approaches to get multilevel scaling.

Based on the type of video, and the choice of scalability order, multilevel scaling can provide different residual errors [24]. E.g., when a surveillance video is captured by the static indoor camera (say, in a shopping mall or banks), it may be meaningful to move first on spatial scalability, followed by temporal scalability. Most of the information across the scenes stays static, hence missing the temporal aspect, does not affect the SNR values. For such videos, it may be more important to get detailed features of objects, than a smooth motion for the mobile objects. On the other hand, for entertainment-focused video recording from a moving vehicle or for a sports event, it is more advisable to have temporal enhancements at lower layers, while higher layers may bring in spatial enhancements. If we delay temporal enhancement for such contents, the viewer may find the video too jittery to watch.

2.7. Scalable video packaging and communication

In this thesis, we explore video communication for opportunistic DTN; hence, we are not covering the topics for regular Internet-based applications or ad-hoc networks. Lindeberg et al. [25] provide a good survey on these aspects. In the following section, we provide a brief context for video transmission and explore alternatives for opportunistic networks.

We can broadly classify video communication into three categories based on the demand of end-to-end interaction and delay – 1) live-interactive transfer with round-trip delays below a second, 2) live or pseudo-live communication based on application needs, and 3) transmission of pre-recorded content. For the first two categories, on regular Internet, mostly RTP/UDP based approach is used [26]. For both AVC and HEVC (and by that virtue SVC and SHVC), encoded content is packed within network adaptation layer

(NAL) unit. Encoder embeds the parameter used for encoding as well as encoded pictures within the NAL units (NALU). During communication, we map these NAL units on RTP/UDP packets. Transmitters try to ensure that the UDP packet size is small enough to avoid fragmentation when the transmission is over the Internet. Each UDP packet contains one or more NAL units of an SVC layer. Actual UDP communication may be multicast (one-to-many) or unicast (one-to-one). Transmission of RTP/UDP content can use different communication channels (ports) for NAL units of each SVC layer. Receivers can use approaches like RSVP [26] to detect available bandwidth and get access to as many SVC layers as possible, starting with a channel for a base layer. In the absence of RSVP, lower layer packets may be assigned higher priority so that these packets encounter a minimal drop and have less delay in the network. To achieve such behavior from the nodes on network, special values in IP header field may be used, or edge routers may have more involved approach using MPLS, etc. The source, using feedback received from clients, adapts the encoder [26]. Receiver implements the network behavior detection while the source does the control video encoding.

When the live transmission is not of interest, video can be broken into chunks and stored. These chunks are subsequently transmitted to regular Internet applications. E.g., DASH [19] stores video in chunks of few seconds at different bit-rates and the transfer happens from the server to the client using TCP. Client downloads and buffers newer chunks, while it continues to decode and playout prior chunks. Since the server and the clients have round-trip delays below a second, the latter can choose lower or higher quality for subsequent chunks, depending on the time taken to download prior chunks. In this case, the client does congestion detection and adaptation. Note that the default DASH approach is different from SVC since the source creates different chunks for different bit-rates and the client chooses to download only that chunk it deems fit, as per its adaptation state. Grafl et al. in [27] explore an extension of DASH involving SVC.

As compared to legacy wired networks with static nodes, there is a higher impact to media flow for wireless networks because of interference, lower bandwidth, and node mobility. Wireless networks, especially Ad-hoc networks (including Manets and Vanets) may use link layer optimizations and other network coding approaches [25] to ensure a higher quality of base layers. Further optimization can be targeted based on deployment behavior. For example, Lu and others in [28] analyze video flow if link delivery probabilities between nodes stay static. On the other extreme of Ad-hoc networks are Vanets where nodes have frequent connections and disconnection. Wu et al. in [2928] have explored VANET for routing of video stream in a dense scenario (more than ten hops and using ten thousand taxi nodes and a radio range of 250 meters). Jason[30] have explored an adaptive layer distribution where the few lower layers of SVC are packaged and transmitted using MDC and measured the impact of loss for such communication across a single hop network.

As we move from Ad-hoc networks to delay-tolerant opportunistic networks, we may have delays of the order of hours. Hence, adaptation control from the receiver may be too late, and source may end up underutilizing or congesting the network. Lack of feedback (e.g., acknowledgment getting lost) can further amplify such issues. Instead, it makes sense for the source to estimate the network behavior using acknowledgments that it receives, and adapt to subsequent transmissions. The adaptation needs to ensure that it is neither too fast while reducing the SVC layers, nor too slow in adapting to increased load on the network behavior.

2.8. Video over DTN

Prior work of video streaming over DTN has been mainly in the context of distributing content or streaming video over oracle-based delay tolerant networks. In some cases, video streaming is done using DTN enabled VANETs. Lenas et al. [31] have proposed a bundle streaming service using single copy forwarding. They also verified the proposal for single-copy routing on the real-world experimental test bed. Morgenroth[32], Blanchet[33] and Cabrero [34] have demonstrated media streaming capabilities in a controlled environment with a limited number of nodes. Raffelsberger [35] adapted DTN to HTTP for video streaming in a simulated scenario of an explosion in a chemical facility. Such approaches do not fit well for opportunistic networks where contacts are intermittent, and count of nodes may vary. In [36], Pan et al. have explored routing hierarchical video in opportunistic networks. They have optimized video streaming, by aligning buffer management of intermediate nodes. However, optimizing the DTN forwarding strategy for hierarchical media flow puts other applications at a disadvantage.

MobEyes [8] explores the use of cameras on moving vehicles. However, it summarizes findings from video and transmits only the summary results (e.g., number-plate of the cars in the video) over DTN or VANET; hence cannot be considered an example for video streaming.

Sandelescu[37] et al. in their work (Guaranteed capacity bound) have compared the performance of a video like flow between a stationary source and a destination (e.g., public library and airport, in one of the use cases) for bandwidth estimation to periodically send video bundles based on the estimates of available capacity. In [37], continuous media flow and video quality measurements are not included. The experiments relied on data from prior round of transmission, with a large time gap between the transmission rounds (e.g., transmission at 06:00, 14:00 and 00:00 hours). Moreover, the work focussed on demonstrating the accuracy of bandwidth and delay estimation, rather than proposing a complete system for streaming video chunks.

Klaghstan et al. [38-40] have analyzed the performance of SVC using multiple layers. In [38] they mapped SVC layers to separate bundles, while in [39,40] they attempt media flow using NAL units. In [38], they have analyzed the delay-and-delivery ratio based on single transmission over different network scenarios and observed that the base layer should have higher copy counts, and the higher enhancement layers should have lower copy count. When using NALU [39,40] they have implemented media-aware network elements (MANE) that combined multiple NALU into a DTN bundle and exchanged them during node encounters. Size of the bundle was estimated based on the contact duration spread observed by a node. All nodes transmit base layer NALU in a single bundle without forcing its fitting into optimum bundle size. In [40] Klaghstan also explored improving the delivered quality further by implementing pull requests from the receiver to fill intermediate SVC layers. If the receiver has got a significant number of NALU for higher SVC layer but missed some NALU for intermediate layer(s), it solicits retransmission from nodes in the network. The nodes in vicinity created additional copies of NALU based on solicitation requests. To avoid overload on network, solicitation is triggered only after a threshold. The threshold is determined based on expected play out time and percentage of NALU received for intermediate layers. All the three works [38, 39, 40], ran the simulation multiple times, but each run involved only one burst of video transmission.

When transmitting a series of video bundles, some of the DTN bundles from the source may not reach the destination because of multiple reasons. One of the most commonly studied reasons for delivery delay or failure is related to congestion or excessive loads on some nodes [11]. If some nodes run out of space when receiving a bundle to be forwarded, they may drop older bundles. Even if large buffers are available, the contact duration between nodes may constrain and limit the exchange between nodes. In some cases, delay and drops may occur because of the deployment scenario, e.g., location and mobility patterns of source and destination nodes are such that they have very few contacts with other nodes.

To reduce congestion, avoidance algorithms may refuse newer bundles, discard carried messages or control replication within the network. [11]. These congestion avoidance algorithms may be transparent to the DTN routing algorithm or update the DTN routing algorithm for better congestion control. To keep the network model simple, we have not analyzed this approach as congestion avoidance on network elements can cause complicated interaction between end-to-end control and network congestion avoidance [25].

3. VIDEO COMMUNICATION OVER OPP-NETS

Overview: It is extremely difficult to transmit large data like videos in opportunistic networks, as the delay may be of the order of hours and we may lose part of the information. The primary challenge is to generate video traffic in a manner that the resources are used optimally, even though feedback may be delayed or lost. To ensure that the network is not under-utilized and, at the same time, the source does not overload the network, we study a novel system that transmits chunks of video compressed and packetized using scalable video coding (SVC). The source uses partial information from prior communication to estimate the network bandwidth and delay. Subsequently, we use the estimate to adapt the transmission of next video chunk.

The scalable nature of SVC allows improving quality if the current conditions of Opp-Net allow the same. But, the video at SVC receiver can suffer from resource wastage, when some of the lower SVC layers are lost or delayed, and it cannot successfully decode the received higher SVC layers. Multiple approaches have been studied to minimize wastage. In end-to-end based approach, the source tries to control the attributes of payload it injects in Opp-Nets. We also study the extensions to Spray-and-Wait (SNW) routing protocol, to ensure better delivery video quality and to lower wastage.

Through simulation, using both real-world traces and synthetic mobility models, we evaluate the performance of the proposed solutions under multiple scenarios. Amongst the extension to SNW routing, preference of higher copy-count shows the maximum improvement for SVC-based video communication. Experimental results show that adaptive control reduces overall delay and minimizes wastage while improving the quality of video at the receiver. Adaptive SVC transmissions can decode almost thrice the bursts compared to non-SVC transmission.

3.1. Introduction

To recap, in opportunistic networks, the destination may only receive parts of multimedia content. Even for the properly delivered multimedia content, some of the acknowledgments may be lost. The round trip times may be of the order of hours; hence, the received acknowledgments convey the behavior as observed a few hours ago. Further, acknowledgments convey information only about that part of the network, over which successful delivery of content and acknowledgment has happened. Such path limitations mean both the source and the destination can get only a partial view of the network. We have to use this partial view of network, and adapt subsequent transmissions from the source, to maintain video quality without overloading the network.

Clark et al. in [15], observed that end-to-end (E2E) principle has contributed to Internet being very successful today. The E2E principle assumes that most of the intelligence is on the end hosts and the network carrying payload is as simple as possible. Since the end hosts are easier to program, innovative applications can be rapidly developed. Hence we attempt to keep the video stream adaptation decoupled from network. Interestingly, future Internet architecture attempts to have more intelligence within the network. Hence, we also explore the benefits that can be achieved when other nodes in the network are aware of the media flow.

We make following three primary contributions:

a) Design and develop a source based adaptation algorithm for effectively transferring video chunks using end-to-end principle [15] over DTN. To the best of our knowledge, there are no prior proposals for adaptive SVC media flow over DTN networks.

b) Explore the impact of extensions to SNW routing protocols on SVC media flow.

c) Analyse the impact of different scenarios (including mobility pattern, load and node density) on SVC media flows.

Additionally, we experiment with the adaptation of individual components to quantify their benefits in isolation. We also analyze the impact of prior proposed extensions to SNW for SVC communication (deletion of messages on acknowledgment and preference to higher copy counts while relaying messages).

3.2. Designing the Adaptive system:

We adapt three aspects of video payload for subsequent transmission – SVC operating points (i.e., number of SVC layers transmitted by source), replication count for different layers, and time-to-live (TTL) for different layers. Optionally, we can redistribute the savings from the adaptation, by increasing the copy count of transmitted layers. The adaptation is purely on the end host and does not rely on a modification of the routing protocols.

We call this system SORT (**S**NW based adaptive video transmission using **O**perating point, **R**eplication count and **T**ime-to-live). For scenarios where it is feasible to have media-aware network elements (MANE) [47], we implement layer awareness to SNW routing for all nodes and analyze its performance.

Based on analysis of media flow (both scalable and non-scalable) across different scenarios, we identified the following goals as ideal for media transmission on opportunistic networks:

a. Aligning with end-to-end principles [15]: this implies that optimization of media flow should not require explicit support from DTN routing layer.

b. Using optimizations from network (if available) to improve media flow performance, as long as it does not impact other (non-media flow) communication.

As mentioned in Chapter 2, multiple DTN routing protocols exist. Some of them are quite complicated and may cause unpredictable feedback with end-to-end adaptation. Hence, we chose binary SNW for its simplicity and explored adaptation by only involving source and destination nodes. The only extension to SNW for intermediate node was to accommodate cumulative acknowledgment, wherein, on receipt of the newer acknowledgment, we discard the older one.

Another key decision point is mapping of communication units to SVC layers or NAL units. Some of the prior works [39,40] have directly mapped NAL units to payload for transmission over DTN. This required changes to all DTN nodes so that they pack/unpack the NAL units into bundles as needed. Since each video frame will generate at least one NAL unit (besides multiple SVC layers and compression parameters that will add more payloads), such mapping will create a significant overhead for continuous media streaming. The communication on DTN will be pseudo-live (delay of the order of hours). Hence we chose to create a burst of DTN bundles every few minutes. The source will transmit each SVC layer as a separate bundle, without the need for any changes on other nodes. This mapping is similar to the communication approach suggested in [37, 38]. We additionally include logic for adaptation from source for continuous media flow.

The DTN bundle specification [48] provides bundle priority as bulk, normal and expedited. In our work, we have not proposed usage of this field – since the video capture is done opportunistically and may use normal or bulk mode, leaving expedited mode for other important communications. Any improvements from the usage of such features will be complementary to our proposal.

Figure 3.1 below provides an overview of changes proposed in this work. Note that across different scenarios, the source (S), destination (D) as well as multiple intermediate nodes (Ni, Nj…) may be mobile. End-to-end adaptation by SORT only involves application layer at the source and destination nodes. Destination generates the cumulative acknowledgment. All other nodes can have regular SNW implementation. In the latter part of this section, we explore extensions of SNW, where nodes give preference to higher copy count (H), delete delivered messages (D) and MANE (M). Figure 3.1 captures these optional extensions as HDM.

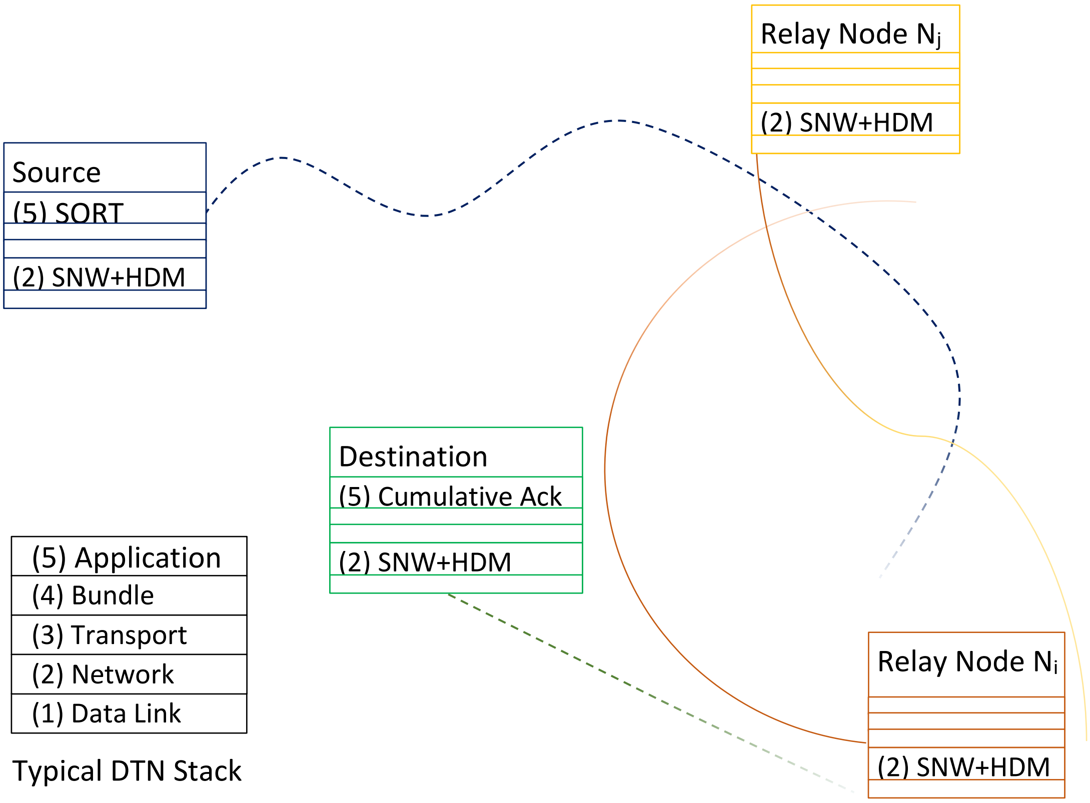


Figure 3.1 DTN stack modifications for different nodes and their mobility patterns

The source (Src) generates a burst of DTN bundles carrying different SVC layers. Intermediate nodes (N), relay / carry these bundles to the destination (Dst). On receipt of bundles at the destination, a cumulative acknowledgment is generated capturing time of receipt of the last few SVC bursts. Figure 3.2 depicts such a media flow over time. Table 3.1 shows an example of the metadata in cumulative acknowledgment.

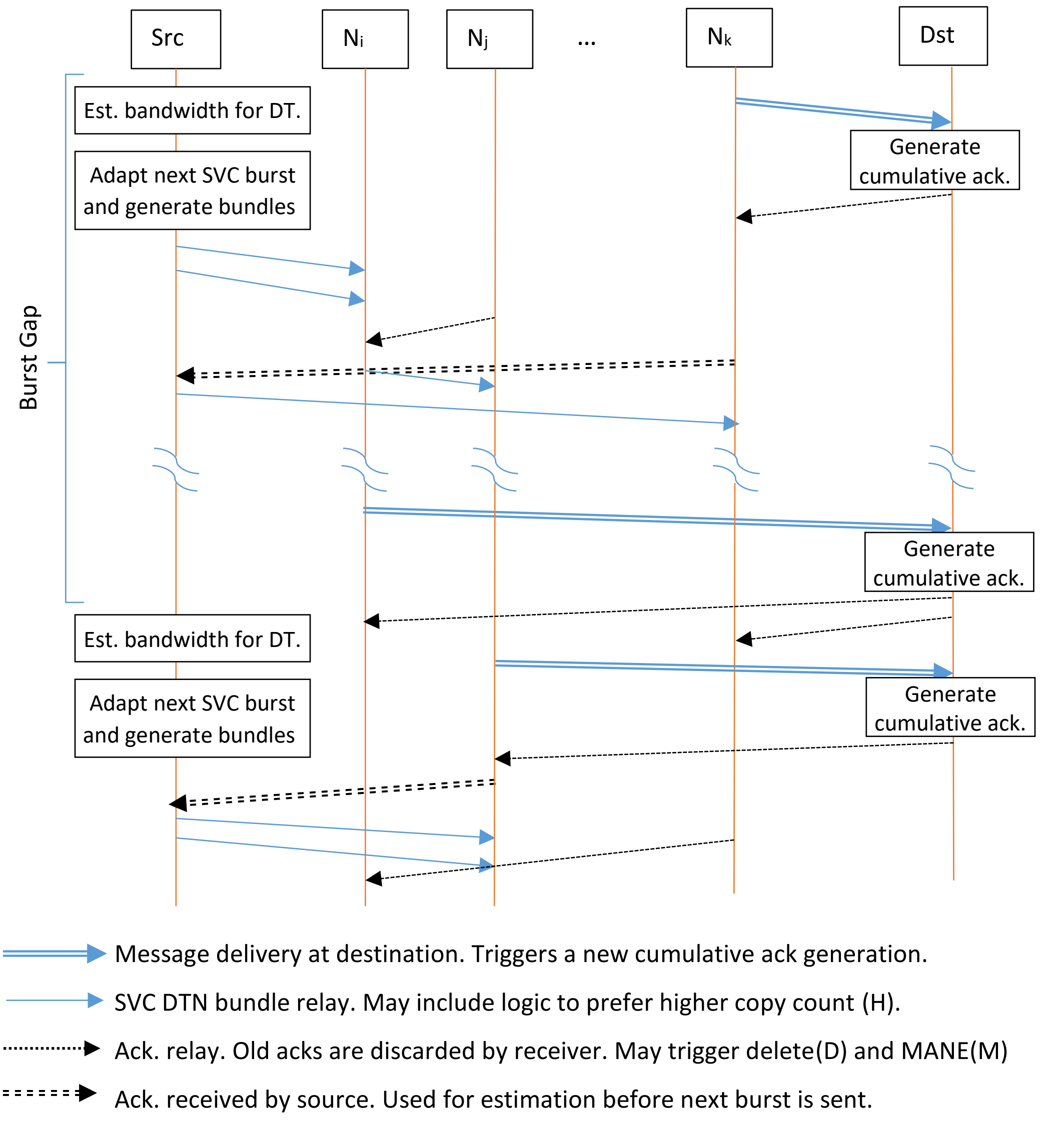

Src
N_i
N_j
...
N_k
Dst
Burst Gap
Est. bandwidth for DT.
Adapt next SVC burst and generate bundles
Generate cumulative ack.
Generate cumulative ack.
Est. bandwidth for DT.
Adapt next SVC burst and generate bundles
Generate cumulative ack.
Message delivery at destination. Triggers a new cumulative ack generation.
SVC DTN bundle relay. May include logic to prefer higher copy count (H).
Ack. relay. Old acks are discarded by receiver. May trigger delete(D) and MANE(M)
Ack. received by source. Used for estimation before next burst is sent.

Figure 3.2 A temporal illustration of various components in the system

Note that Figure 3.2 does not capture the mobility of nodes. The source node adapts the next burst based on acknowledgments it has received. Intermediate nodes relay the bundle. If the SNW extensions for higher copy count (H) are enabled, choice of the relayed bundle is affected by the value of L. As and when destination node gets the SVC bundle;

it generates a new cumulative ack. Nodes share the cumulative acks. On receiving a new ack, nodes may trigger SNW extensions for MANE(M) and delete(D), if they are enabled.

In this section, we first discuss video packaging and transmission bursts of bundles with different SVC layers from the source. Subsequently, we provide the logic of cumulative acknowledgment. Then we present an algorithm for estimation of congestion at the source. Based on the estimates for congestion, we discuss the adaptation from source after that. Finally, we cover the optional extensions to SNW.

3.2.1. *Video packaging and transmission*

For communicating video over delay tolerant opportunistic networks, the video is captured for a few seconds to a few minutes and subsequently compressed using SVC. Each of the SVC layers is transmitted using different max-copy-count threshold (L). Similar to [38], the base layer gets highest counts, and we gradually reduce the copy counts for subsequent enhancement layers. The source generates a burst of DTN bundles every few minutes. Typically, the inter-burst-gap will be the same as the duration of the video capture.

The receiver (destination) will have some delay target, for playing back the received video, from the time the source sends it. The source will set the time-to-live (TTL) at a value much higher than the delay target (e.g., twice the delay target in our case). TTL values in DTN ensure that, in the absence of acknowledgments, the intermediate nodes do not continue to cache the data forever. They delete the bundles whose TTL has expired.

3.2.2. *Cumulative acknowledgment*

DTN routing has an optional feature to send an acknowledgment. Acknowledgments are small bundles that may be flooded on the network or communicated using the same routing protocol as the one carrying the application payload. These acknowledgments can be used to delete delivered messages if the feature is enabled [11, 36, 49]. While earlier work used separate bundles for acknowledgment, we propose usage of cumulative ack to increase the chance of delivery of acknowledgment to the source of media flow.

Cumulative acknowledgment includes delivery delay for the last few bundles in the media flow between the source and destination pair. Since the acknowledgments are cumulative, missing some of the acknowledgments will not significantly impact estimation at the source.

Table 3.1 provides a simplified representation of metadata within the cumulative ack. Note that the cumulative acknowledgment differs significantly from TCP cumulative acknowledgments. It allows holes to exist in the sequence; there is no expectation of retransmission of missing bundles, and it includes delivery delays for each received bundle. For example, in Table 3.1, for burst 41, EL1 had a delay of 73 minutes, while EL2 was not received when the ack was generated. Using simple bitmaps and rounding off receipt delay to closest minutes, we can communicate cumulative acknowledgment for the previous four hundred bursts in less than one kilobyte. Application of other lossless compression schemes can reduce this data further. The design of optimal compression scheme for these acknowledgments is outside the scope of this thesis.

Table 3.1 Metadata in cumulative acknowledgment from the destination

<table>
<tr><td colspan="7">Source of ack: 17 {identity of Dst}</td></tr>
<tr><td colspan="7">Destination of ack: 44 {identity of Src}</td></tr>
<tr><td colspan="7">Cumulative ack seq. number: 109</td></tr>
<tr><td colspan="7">Oldest burst id: 41</td></tr>
<tr><td rowspan="5">Entries for delay in</td><td>Burst Id</td><td>BL</td><td>EL1</td><td>EL2</td><td>…</td><td>ELN</td></tr>
<tr><td>41</td><td>74</td><td>73</td><td>-1</td><td>…</td><td>-1</td></tr>
<tr><td>42</td><td>94</td><td>-1</td><td>33</td><td>…</td><td>-1</td></tr>
<tr><td>..</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>90</td><td>76</td><td>-1</td><td>77</td><td>…</td><td>…</td></tr>
</table>

On receipt of a new acknowledgment bundle (with higher cumulative ack seq. number, for the same source-destination pair), the intermediate nodes discard the earlier acknowledgment. SNW routing first attempts to deliver the bundles to the destination and then exchanges acknowledgments before forwarding other bundles. Note that rather than implicitly sharing acknowledgments during contact, we explicitly share the acknowledgments, making the algorithm much more practical.

3.2.3. *Bandwidth and delay estimation*

We have designed the following algorithm to estimate likely congestion between source and destination in the DTN network. Note that the computation of congestion estimate uses the data only about SVC layers and bursts of transmission between the source and destination pairs. It is not an estimator of congestion in the overall DTN network.

Table 3.2 Algorithmic constant used in SORT and HDM

Identifier	Description / Value
DT	Delay target - ideally playback at the destination is for a burst sent at (Tnow-DT)
BG	Burst gap – gap between each SVC burst
RMDP	Ratio of missed bundles penalty to delayed bundles penalty (value = 9)
EstChoice	Choice between average, minimum, maximum of the two estimates
MxLayers	Maximum number of SVC layers.
MxTtlDrop	Maximum factor by which TTL for higher layer is reduced (default = 0.2)
ActualTtl	Default value of TTL

Table 3.2 covers details of constants used in the algorithms. Application running at the source can set it, based on the requirements of deployment Note that the values used for these constants are based on limited experiments across different scenarios. We have not attempted to identify optimal value for them as we found that, for different scenarios, optimum values could differ quite a lot.

The estimate of congestion using NumSample involves counting the number of delayed or missed base layers and enhancement layers. Since base layer is an absolute must for decoding at the destination, it is given half the weight in the estimate (line 5 of Algo. 3.1). All enhancement layers contribute to the other half. Moreover, missed bundles are much worse than delayed bundles. Hence, a higher weight is assigned to missed bundles using RMDP. Note that division by “2 * (1 + RMDP)”, ensures that the estimate is range-bound between 0 to 100.

Even the cumulative acknowledgments may be delayed or lost; hence, sometimes the state may be wrongly represented at the source. Moreover, network behavior can vary over time (because of node mobility). Therefore, two intervals (DT and 2*DT) are chosen to estimate the congestion. Choice of average/min/max (EstChoice) of the two estimate is driven based on whether the SVC media flow can be aggressive on resource usage or not. Algo. 3.2 combines the two estimates based on the value of EstChoice at the application level.

Algorithm 3.1 getStaticticalEstimate(numSamples)

```
Tnow = getPresentTime();
mxBurst = (Tnow-DT) / BG;
mnBurst = maxBurst – numSamples;
Using prior received acknowledgments for [mnBurst to mxBurst ]
        i. cBLmsd = count of base layer missed (no-ack);
        ii. cBLdlyd = count of base layer delayed (delay > DT);
        iii. cELmsd = count of enhancement layer missed (no-ack);
        iv. cELdlyd = count of enhancement layer delayed (delay > DT);
        v. tELtx = total count of enhancement layers transmitted;
newEst = $100 * \left\{\frac{\mathrm{RMDP} * \mathrm{cBLmsd} + \mathrm{cBLdlyd}}{\mathrm{numSamples}} + \frac{\mathrm{RMDP} * \mathrm{cELmsd} + \mathrm{cELdlyd}}{\mathrm{tELtx}}\right\} / \{2 * (1 + RMDP)\}$ ;
Return newEst;
```

Algorithm 3.2 estimateCurrentState ()

```
sSamples = DT/BG;
shortTerm = getStaticticalEstimate(sSamples);
lSamples = (2 * DT) / BG;
longTerm = getStaticticalEstimate(lSamples);
avg = (shortTerm + longTerm)/2;
min = Min(shortTerm, longTerm);
max = Max(shortTerm, longTerm);
finalEst = selectEstimate(EstChoice, avg, min, max) ;
Tnow = getPresentTime();
Return finalEst;
```

3.2.4. *Source adaptation based on state estimation*

Based on the congestion estimate, the source may infer one of the following states: No Loss or Delay, Occasional delivery beyond delay target, Sustained loss at delay target or Sustained loss beyond delay target (at TTL). Fig. 4.3 captures these states and adaptation values associated with each of them. Based on experiments, we chose these adaptation values a reasonable trade-off across different scenarios. The choice ensures that end-to-end adaptation reduces the load on network for simple SNW deployments while giving significantly better results for decoded video quality. We discuss usage of adaptation values (OP_ratio, RR_Ratio, and TTL_ratio) in the context of Algo. 3.3.

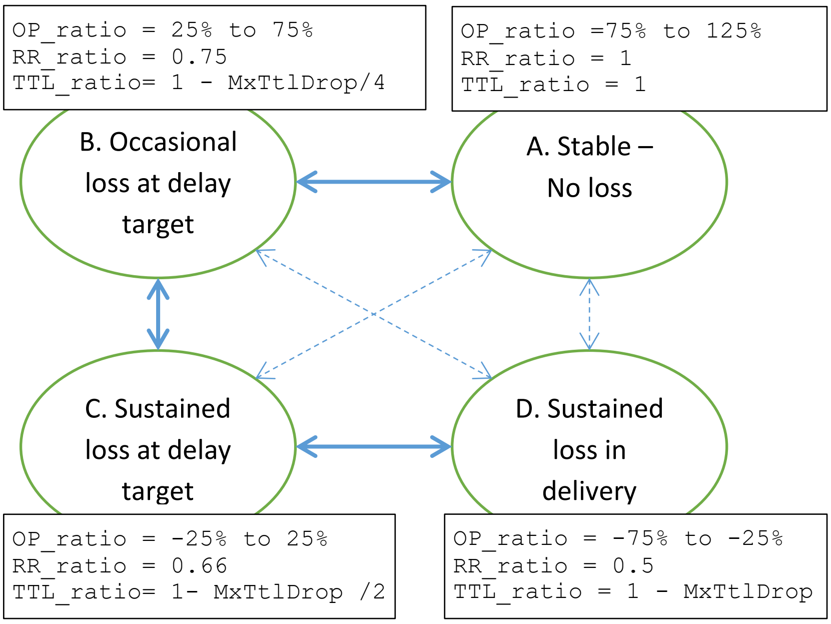


Figure 3.3 DTN stack and its modifications for different nodes

In general, transitions should happen along the solid edges of Figure 3.3, but in the presence of sudden changes in the network conditions (say bursts between other nodes or extreme mobility of source or destination), alternative state transitions along the dashed lines cannot be ruled out. The finalEst from Algo. 3.2 is mapped to State A [0,25]; State B (25,50]; State C (50,75] and State D (75,100].

We cannot apply conventional TCP/IP like flow and congestion control to opportunistic Video transmission, on three accounts. First of all, as observed in regular Internet multimedia communication, Video transfer can be loss tolerant but would expect reasonable bounds on delay. Secondly, unlike TCP transfer, some parts of the video can be discarded by the source when it observes severe congestion. Finally, based on congestion, it is possible to control replication overheads in DTN. We transmit the few SVC bursts without any adaptation. Subsequently, based on congestion estimate, the source adapts future bursts. Since we are relying on SNW over DTN, we have the following three levers to adapt:

a. O - SVC **O**perating point - In the presence of severe congestion, the source does not transmit one or more of the higher layers. Less number of bundles will imply smaller load on the network. OP_ratio from Figure 3.3 is used to control this value.

b. R - **R**eplication counts of bundles – The source reduces the replication count for higher layers to ensure lower load on the network. RR_ratio controls this value.

c. T - **T**TL values – The time-to-live for higher layers can be reduced to ensure that they spend less time in the buffers. When buffers are full, and some messages need to be deleted for space, smaller TTL values for higher layers will ensure that nodes running out of storage, delete bundles for higher layers before deleting lower layer bundles. This adaptation uses TTL_ratio from the corresponding state in Figure 3.3.

Algorithm 3.3 adaptNextBurst ()

```
Initial values: eWMA_PathLen = MaxLayers
est = estimateCurrentState();
eWMA_PathLen = (1-$\alpha$) * eWMA_PathLen + $\alpha$ * MxLayers * OP_ratio;
ensure eWMA_PathLen is below MaxLayers and above 1 (base layer).;
for (layer = 0; layer < MxLayers; layer++) do
    if(eWMA_PathLen < layer) break;
    ttl[layer] = ActualTtl * (TTL_ratio) ^ layer;
    if( layer > 0 and layer >= eWMA_PathLen -2) then
        copyCount[layer] = RR_ratio * cc[layer];
done
if ReDist and (higher layers removed or copyCount or ttl reduced) then
    increment the copy counts for lower layers.
```

Furthermore, when the source adapts O/R/T, it can measure the amount of traffic load being taken off the network, and redistribute (ReDist) the same to lower layers. Typically, lower layers of SVC are smaller in size. Hence ReDist can increase the copy counts for base layer and lower enhancement layers, thus improving the delivery rate and reducing delivery delay for them.

To ensure that changes in path length are gradual, an exponential weighted moving average is done for operating point using α as 0.2 as captured in line 2 of Algo. 3.3.

Assume that five SVC layers are used in transmission (MaxLayers=5), and prior eWMA_PathLen is at 4. Let us consider the impact of Algo. 3.3 when the estimate is at 98 (state A, OP_ratio = 121%) and 35 (state C, OP_ratio = -5%). For state A, the new value for eWMA_PathLen will increase to 4.41 (0.8 * 4 + 0.2 *5 * 1.21) and hence logic on line 5 will transmit four layers. There is no impact to TTL, and copy count in State A since both TTL_ratio on line 6, and RR_ratio on line 8 are at 1. On the other hand, if we are in state C, the eWMA_PathLen will reduce to 3.15 (0.8 * 4 - 0.2 * 0.5 * 0.05). Thus, it sends only three layers (BL + EL1, EL2). Besides, for enhancement layers, copyCount(L) and ttl will reduce. Value for ttl[1] will drop to 90% while it will be 81% for ttl[2]. Similarly, copyCount will reduce to 66% for enhancement layer 1 and 2.

3.2.5. *SNW extensions*

While SORT adapts SVC media flow without changes on intermediate nodes, we explored alternatives to improve the delivery of content, by making the following three changes to SNW routing. As mentioned before, similar extensions have been proposed in the past [14, 38, 40], but to the best of our knowledge, there is no prior study of their impact on hierarchical content like SVC media flow. The first two extensions are quite generic and will not give undue resource allocation to SVC media flow.

1) H - prefer **h**igher copy count: When two nodes are in contact and need to relay bundles, vanilla SNW proposes a random order of relaying, or relaying based on earlier packet first. For SVC, we know that lower layers will have higher copy counts. Hence we extend SNW to share bundles, based on the count of replica (L) held by a node. To relay bundles, it starts with the bundle for which it has maximum copy count.

2) D - **d**elete on acknowledgment: On receiving a cumulative acknowledgment, intermediate nodes delete the existing bundles for the delivered SVC layers.

3) M – **m**edia-aware network elements: Source-based adaptation is a push-based model and does not allow the destination to make pull requests. In case some of the higher layer bundles are received at the destination with missing intermediate layers, it makes sense to increase the counts for missing intermediate layers to ensure that they also are delivered. Only after delivery of missing intermediate layers, prior received higher can be used for decoding. When MANE extension is enabled, the nodes periodically check acknowledgments to identify the gaps. MANE uses a timer-based threshold and doubles the replication count (L) for bundles matching missing intermediate layers.

When MANE is enabled, nodes in the network create extra copies for intermediate layers; when they detect delivery of higher layers and a miss for one or more lower layers, for the same SVC burst. This increases the likelihood of the delivery of missing layer(s), improving the quality while reducing wastage from the decoder's perspective. Since MANE creates provisions for responding to gaps within acknowledgments by creating extra copies for SVC media flow on the network, it deviates from the network-neutrality principles and also breaks the end-to-end paradigm [15].

3.3. Experimental setup

For our experiments, we have simulated flow of SVC bursts in different network topologies. Based on bundles received at the destination, we have measured the peak-signal-to-noise-ratio (PSNR) for various delay targets. We map the PSNR values to mean-opinion-score (MOS) for ease of analysis.

3.3.1. *Input video characteristics*

For our experiments, we used publicly available test video data Highway [50] with 2000 frames at 30 frames per second and CIF resolution (352x288) for generating video load. JSVM is used to create nine-layer SVC content with layer-specific details as captured below. Further x264 (0.148.2643) is used to create a non-scalable video with a target bitrate of 100 kbps. The resultant file is 836 KB in size with luminance PSNR of 28.7159.

To understand the impact from different SVC ordering [24], we also experimented with seven-layer SVC where we first went for temporal scaling at QCIF (BL + 4 temporal EL), and then did spatial and quality scaling. Across simulation runs, the trends for seven-layer plots were similar to those for nine layers, and hence they are not included in this work.

Note that we have used only luminance(Y) PSNR as generated by the PSNRStatic tool of JSVM [51]. The chrominance PSNR values (U, V) were always above 38 dB across SVC layers for this video set.

Table 3.3 Nine Layer SVC operation points for Highway test video sequence

Layer	Size (KB)	Luma PSNR	Type	L at Source
BL	92	27.0608	176x144 at 1.875 fps	32
EL1	212	27.1757	352x288 at 1.875 fps	16
EL2	76	29.1398	352x288 at 3.75 fps	12
EL3	92	31.3958	352x288 at 7.5 fps	10
EL4	108	33.1269	352x288 at 15 fps	8
EL5	132	35.7351	352x288 at 30 fps	7
EL6	556	36.7759	Quality enhancement 1 (36-33)	6
EL7	1056	37.954	Quality enhancement 1 (33-30)	5
EL8	1232	38.7324	Quality enhancement 3 (30-28)	4

3.3.2. *Video decoding and quality measurements*

The receiver may not get all payloads sent by the source. Some of the payloads may be lost or delayed, and will not arrive in time for decoding and playout at the receiver. To compensate for missed higher layers (for SVC), the decoder may use multiple approaches like spatial scaling, repeating frames for temporal scaling, etc. [23]. If none of the layers is received, the last decoded frame can be repeated to compensate for missing parts of the video. Such compensation approaches, improve the user experience, but the played out video can be significantly different from the original content.

One can measure the quality of decoded video using multiple approaches [46]. Average value of PSNR across video frames is one of the most common approaches. PSNR measurements can be automated, when both the original content and received data are available. Mean-opinion-score (MOS) is another metric used in multiple research work. Since MOS involves manual feedback collection from multiple users who watch the video, it is very costly to implement and can be prone to bias if not properly implemented. For automatically determining the received video quality, the PSNR values have been converted to mean opinion score (MOS) using heuristic mapping proposed in [46].

Table 3.4 Heuristic mapping for PSNR to MOS

PSNR (dB)	MOS
>37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
<20	1 (Bad)

In our experiments involving scalable video coding over Opp-Nets, PSNR measurement considers only successfully received base layer and consecutive enhancement layer for each burst. If enhancement layers are missing in the spatial domain, lower resolution decoded frames are scaled to a higher resolution before computing PSNR. Similarly, if temporal frames are missing, we used FFmpeg (2.8.11) to generate the intermediate frames. We use MOS of 1 (bad) for video bursts where the base layer itself is not received.

3.3.3. *Simulation tool*

It is not feasible to create all scenarios in real deployments, especially for opportunistic networks. Moreover, the results in the real world will not be repeatable since even a small change in the environment can cause a different sequence of events. Simulation provides an efficient way to repeat experiments in a similar environment and compare the performance of different algorithmic choices.

ONE [41] is a Java-based simulator for opportunistic networks. Though it lacks support for sophisticated link layer technologies, it has many options for higher layers in DTN stack. It can also monitor the buffer size and energy levels on nodes. Besides, it can use maps and external traces in its simulation runs. The link bandwidth and distance for radio interface are also programmable. The biggest benefit of ONE is that it's an open source and is extensible.

In this thesis, we use three main scenarios in the simulation. Random-way-point (RWP) moves nodes on map paths of Helsinki city at different speeds between key points of interest, using shortest path constrained by the map definitions. Once a node reaches the destination, it chooses the next point of interest randomly and starts moving towards the same. Number of nodes, range of their speed, etc. are controlled using configuration files. Different random number seeds can be used to vary the choice of points of interest and speed for the nodes. The RWP generates contacts without any correlation during simulation.

The second scenario used in the thesis uses working-day-model (WDM) [42] where nodes commute between home and office. Some of the nodes also go out for an evening activity, to emulate shopping events or dinner in a restaurant, etc. This simulation includes support for buses and cars. It divides the area of Helsinki city into four zones (A-D). Combination of zones is created as E, F and G. Zone H covers the complete city area. Nodes (denoting people) belong to different zones and commute within the zones for home, office or evening activities. Further, it supports nodes following shortest path based movement between random locations to simulate taxis, ambulance, police vehicles delivery vans, etc.

The third scenario used real-world traces collected from San-Francisco taxis (SFT) over a period of a month[43]. The traces include GPS locations of around 500 taxies and time of capture. Since some of the locations are wrong captures, as part of our experiments, we processed the traces as mentioned below:

1. Segmented the traces on a week-by-week basis.
2. Removed GPS entries beyond [37°, 38°] on latitude and [-123°,-122°] on longitude.
3. Chose the most active taxis for the week (i.e., taxis that reported the maximum number of GPS locations).
4. Converted the spherical GPS locations to planar maps (in X,Y space). It used Haversine equation for distances between points and after that triangulation for

each point (X_p,Y_p) with (X_{min}, Y_{min}), (X_{max}, Y_{min}) to get the actual locations for simulation.

5. Cleaned up GPS traces by removing entries that reported speeds higher than 70 meters per second (i.e., around 160 mph).

3.3.4. *Extensions to ONE*

The configuration files and the code are under version control as a private repository. Below is the summary of significant extensions made to ONE.

1. NetworkInterface.java and CBRConnection.java classes updated to avoid triggering a disconnect when an interface enters hibernate state. Actual implementation was done as an extension in class AdaptiveOnOffInterface.java.
2. Extended SprayAndWaitRouter.java to multiple subclasses to support a) Discrete Poll, controlled by code; b) Sleep/hibernate logic; c) Other discovery algorithm specific code.
3. We created multiple new report classes.
4. For media flow specific routing, SprayAndWaitRouter.java was updated to support cumulative acknowledgment, and other media flow specific extensions.
5. A new mechanism was implemented to simulate media from between a pair of nodes.

3.3.5. *Simulation settings*

Table 3.5 captures the three different simulation scenarios used in the experiments. Besides, we also used variants of RWP with different node densities, delay targets and burst gap.

The network interface (range and speed), delay target, default TTL and buffer size values are chosen for RWP to align with the settings used in [38]. For SFT and WDM, they were updated to ensure that reasonable communication could happen even without the optimizations proposed. The intent was that for non-SVC video without using the adaptation or SNW extensions, approximately half the transmissions reach the destination within the delay target.

For real-world traces (SFT), the cars move quite frequently, but their paths are different, and they cover a significantly larger geographical area. Hence, the choice of WiFi and similar delay target as RWP suits this model. In case of WDM, the nodes travel from home to office and sometimes to evening activities after work. Since there are large periods where nodes stay static (office in the day, and home at nights) the delay target was taken as 24 hours. Since the map area is large and nodes do not move for a significant period, WiFi like interface is used in WDM scenario to create more contacts.

Table 3.5 Simulation settings

	Random Way Point (RWP)	San Francisco Taxi traces (SFT)	Working Day Model (WDM)
Mobility model for nodes	Shortest Path Map Based [41]	Real world taxi traces [43]	Working day Model [42]
Node Details	50 nodes at pedestrian speeds	50 taxi nodes	2 Buses, 48 nodes
Map details	Helsinki with points of interest	San Francisco and its vicinity	Part of Helsinki (Area A in [42]); 10 offices and 10 meeting spots
Mobility Speed and patterns	0.5 – 1.5 meters per sec.	As per trace data	50% car ownership
Network Interface	10 meters, 2 Mbps	50 meters, 24 Mbps	50 meters, 24 Mbps
Default Delay Target	2 Hours	2 Hours	12 Hours
Default TTL	4 Hours	4 Hours	24 Hours
Default Burst Gap	5 minutes	5 minutes	5 minutes
Number of Bursts	600	600	800
Buffer Size	100 MB for the source, 10 MB for all other nodes	100 MB for the source, 10 MB for all other nodes	600 MB for the source, 60 MB for all other nodes

We ran each simulation fifty times with different random number seeds. For analysis, we skipped the bursts in the beginning and end of simulation runs, since the network is not saturated in these cases. Skipping of initial flows also provides adequate time for adaptation to settle itself. For RWP and SFT, the data analyzed is from bursts 200-299;

while for WDM, the analysis is for bursts 432-719. Usage of 288 samples in WDM ensures that we cover the complete day with the default delay of 5 minutes between each burst. SFT and RWP did not exhibit such daily pattern; hence, 100 samples suffice for them.

The simulations have considered only one media flow over DTN and does not analyze the impact of other communication (including other media flows). Since we have experimented with a variety of network deployments, we expect that in the presence of other communication, the trend for results will be similar, i.e., deterioration of quality or adaptation level changes should follow similar trends. For PSNR measurements, we have only considered consecutively received video bursts.

3.3.6. *Cost for DTN communication*

To identify the savings on cost, one of the common metrics used in DTN applications is the overhead-ratio. This is the ratio of all bundles received and created across nodes, to the number of delivered bundles (see Eq. 3.1). For SVC, given that the size of bundles varies significantly, this metric does not completely capture the associated cost. Hence, we also use cumulative buffer occupancy (CBO) over time in MB-hours for analyzing communication cost. Eq. 3.2 computes CBO, by multiplying the time for which the bundle is in buffers of a node, with the size of the bundle, across all nodes. Note that, for SVC bundles created on source node $M_{\text{Receive time on N}}$ in Eq. 3.2 will be the time of creation of the message.

$$CBO = \sum_{N=1}^{Num-Nodes} \quad \sum_{M=1}^{Num-Messages} (M_{Delete\ time\ on\ N} - M_{Receive\ time\ on\ N}) * M_{size} \ . \ . \ . \ . \ . \ . \quad \text{Eq. 3.1}$$

The total count of delivered bundles normalizes overhead computation. For lightly loaded networks if TTL is large, almost all bundles are delivered. In the absence of MANE, overhead for SNW will be upper-bound to the values of L used by the source application. Similarly, CBO will be upper bounded to the values used by source (for L, TTL) and the number of bundles created.

As network load increases, overheads and CBO can increase in two different patterns. If bundles are deleted because the source (or nodes in the vicinity of source) runs out of space, the “total bundles relayed” as well as “receive to delete time” will be small. Since delivered bundles will be low, we will see higher values for overheads, but CBO will not increase significantly. On the other hand, if the nodes have large buffer capacity and node contact durations are large, (e.g., in case of WDM), the bundles will have a higher count for relayed bundles. They will also stay in buffers longer. For such cases, the delivery rate affects the overhead, but the CBO values will stay high, irrespective of delivery rate.

3.4. Performance evaluation

In this section, we first present the results for RWP scenario for consecutive bundles received and delay in receiving the bundles. Subsequently, we present MOS based results for RWP as well as WDM and SFT run. Next, we analyze the communication costs using overheads and CBO for the three scenarios. After that, we present the impact on HDM and SORT when we generate faster/slower bursts with smaller/larger delay targets. We also present the impact on SVC media flow when the node density is varied. Finally, we analyze the individual contribution from each of the components of SORT and HDM.

3.4.1. *Base RWP scenario – consecutively received layers*

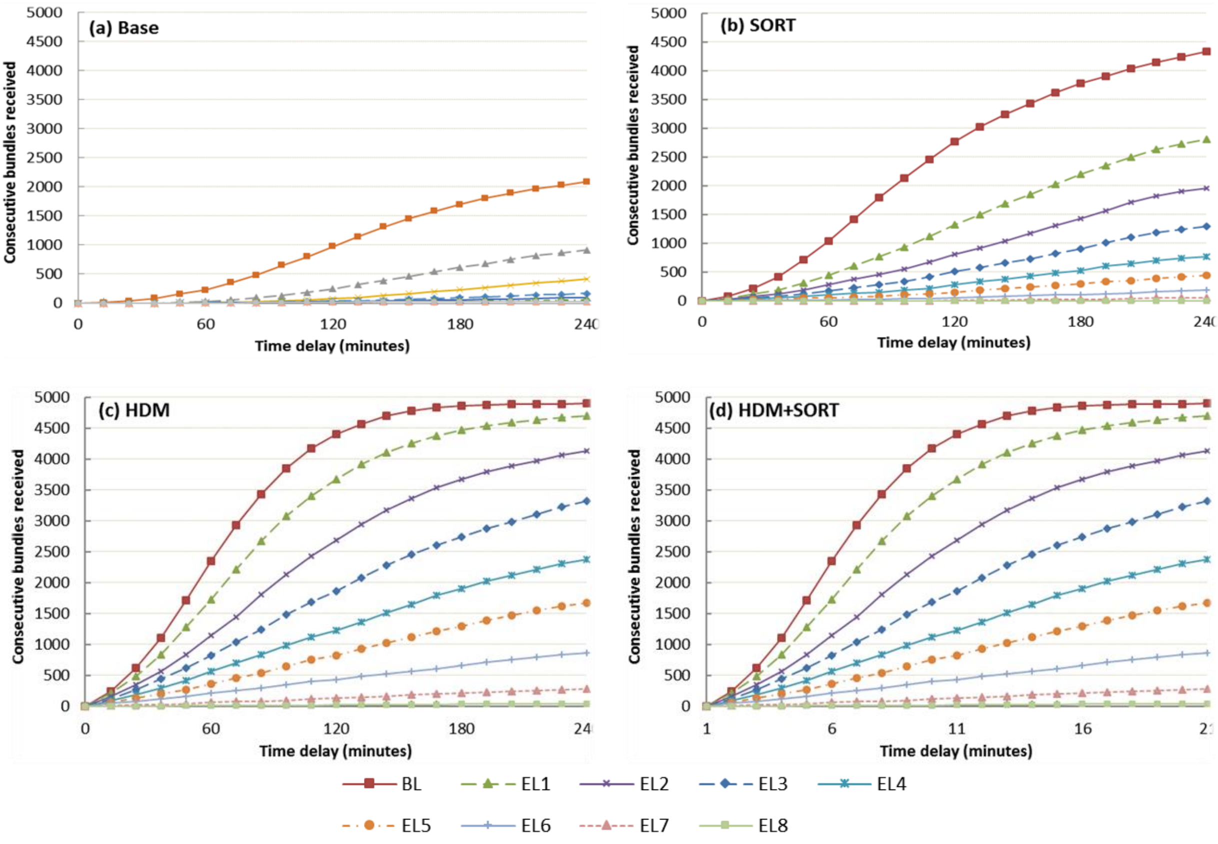


Figure 3.4 SVC Maximum consecutive layer received for RWP base scenario

Figure 3.4 captures the plot for RWP mobility scenario across four different settings. Subplots show the cumulative count for maximum consecutive layer received (for 5000

SVC bursts) on the vertical axis against the delay in delivery on the horizontal axis. First sub-plot, Figure 3.4(a), shows the performance of nine-layer SVC video transmission with SNW implementation supporting only cumulative acknowledgment. At two hours (default delay target), less than 20% of the bursts are decodable at the destination. At this time BL bundles reach the destination for less than 1000 bursts. At the end of 4 hours, 42% of the bursts received at the destination are decodable. Similarly, at delay target, 23 flows (0.5 %) reach EL5 or higher. Less than 51 bursts (1%) reach EL5 or higher with a delay of 4 hours. Note that the plots are cumulative – for example, the plot for EL5 is the sum of all bursts that are decoded till enhancement layer 5 or higher.

Figure 3.4(b) shows the performance when the source adapts the transmission. Higher enhancement layers (EL7 and EL8) still have very low counts. Values for all other layers have improved significantly. BL improved from 20% to 55% at delay target and 42% to 87% for 4 hours. The metrics for EL5 corresponding to two hours is 3% and four hours delay is 9%. At delay target of 2 hours, the destination could decode almost thrice the number of bursts for SORT when compared to the Base scenario.

Figure 3.4 (c) shows the performance when SVC bursts are transmitted with SNW extended to support higher copy count, delete delivered message and MANE. It does not include SORT adaptation. As compared to Figure 3.4 (a), HDM improves the quality of all layers. When compared to SORT, SNW-HDM shows improvements in all the layers. At delay target (120 minutes), BL achieves 88%, and EL5 achieves 16%. At the end of four hours, BL achieves 98% while EL5 achieves 33%.

Combining both HDM and SORT, as captured in Figure 3.4 (d), improves the results at delay target to 89%(BL) and 18%(EL5). The values for four hours delay are 98% for BL, and 32% for EL5. For this experiment, network-based optimizations (HDM) give better results than end-to-end optimization (SORT).

3.4.2. *Quality (MOS) and transmission and storage costs across scenarios*

To properly understand viewing impact of the layers received, Figure 3.5(a), plots the above four scenarios with MOS values for different delay targets. We have also included the baseline plot for non-SVC video and SVC video using constant copy count of eight for all layers (labelled as SVC-Linear). For ease of visualization, plots are from 0.5 * DT (60 minutes) to 2* DT (240 minutes). Figure 3.5 (b) and (c) are the corresponding plots for SFT and WDM scenario.

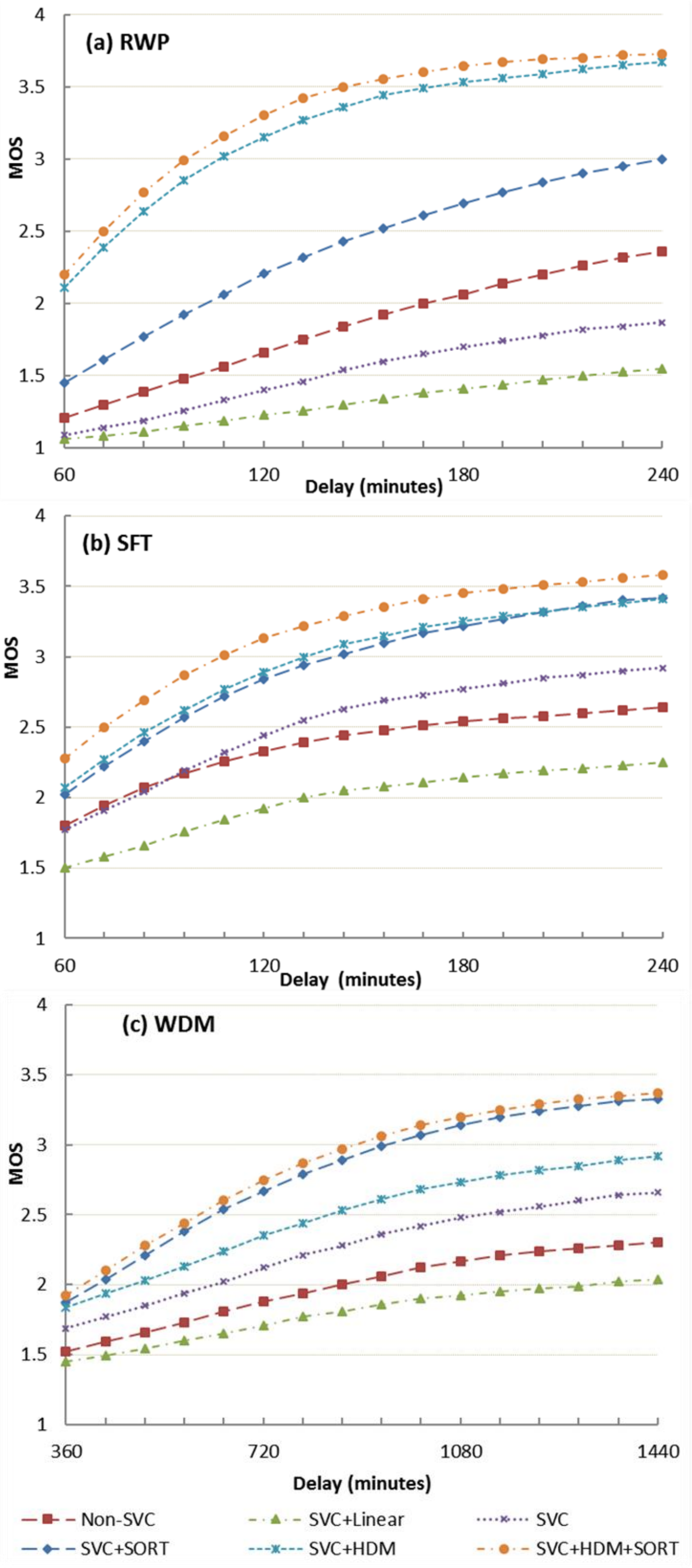


Figure 3.5 Mean opinion scores for RWP, SFT and WDM scenarios

Comparing SVC to SVC+SORT shows that at delay target, across the scenarios, MOS score improved by 0.5 to 0.8. Improvements are much more significant at TTL (twice the delay target) with MOS values improving by 0.7 to 1.1.

Combined HDM and SORT perform the best for all scenarios. HDM outperforms SORT in case of RWP, while it is the other way round for WDM. In case of SFT, both RWP and SORT are very close to each other. Among the non-adaptive versions, SVC-Linear is the worst in all scenarios, as it does not give higher copy count to the base layer.

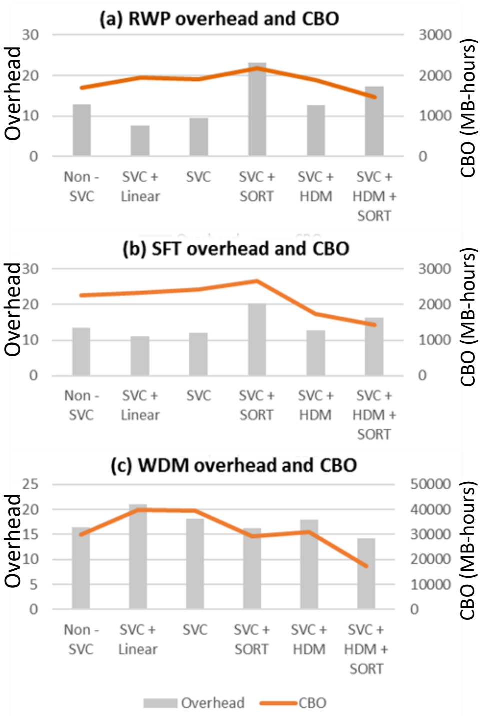


Figure 3.6 Communication and storage costs for RWP, SFT, and WDM

For SFT and WDM, just moving from Non-SVC (single layer, 836 KB content with L=16) to SVC (9 layers with more than 3 MB in total, but with BL of 92 KB and L=32) improves the quality even when HDM and SORT are not enabled.

In Figure 3.6, CBO is lowest when both HDM and SORT are enabled. Individually, SORT has highest costs for RWP and SFT since it does not benefit from delete optimization of HDM. In case of WDM, the costs on CBO are similar for HDM and SORT.

Since overheads only include relay-counts and do not give weight to the bundle size and time occupied by the bundle in buffers, non-SVC, SVC, and SVC-Linear flavors show a lower value on overheads. It should, however, be kept in mind that MOS values for these runs are significantly poorer than those involving HDM or SORT.

3.4.3. *Load and node-count related performance*

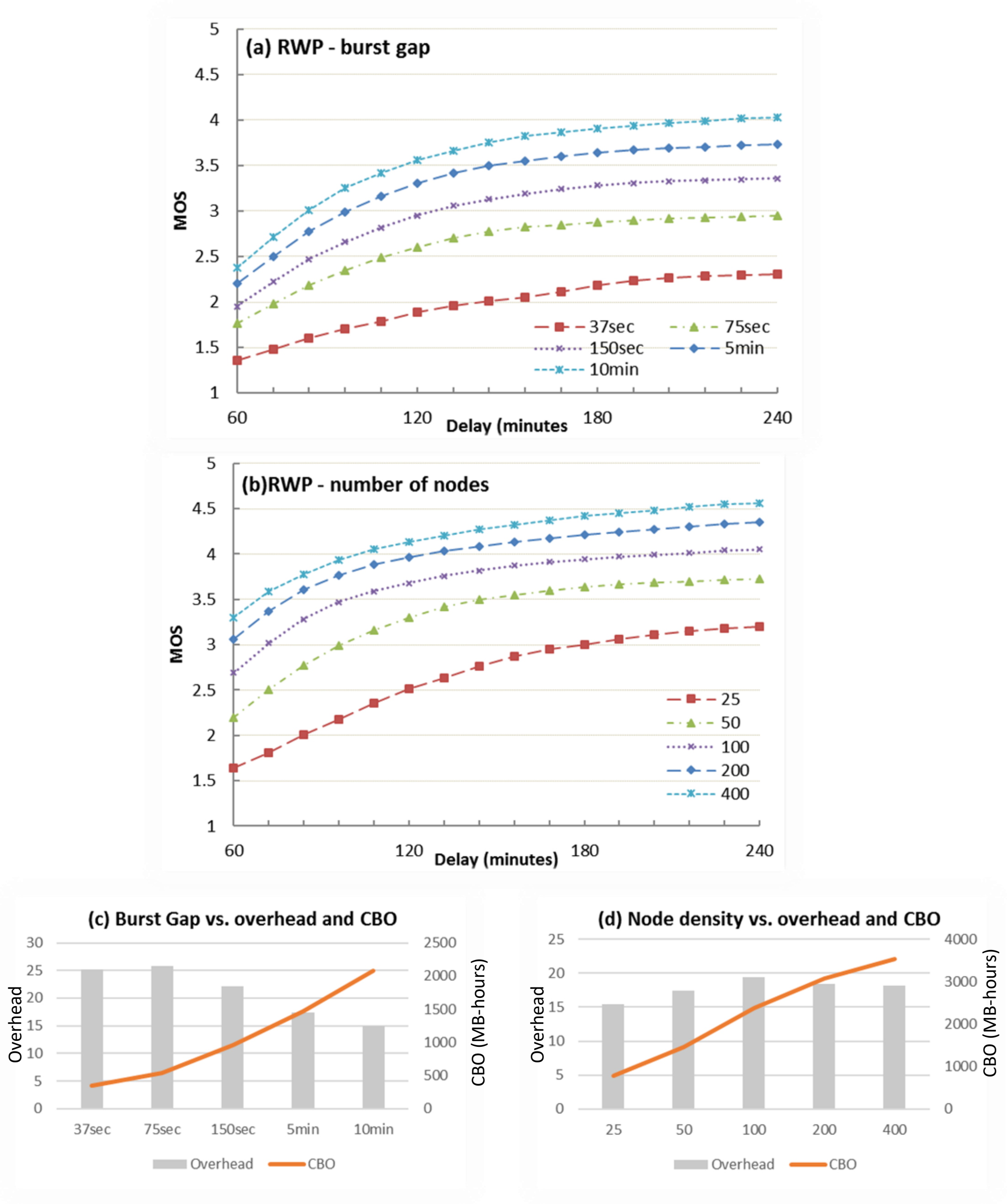


Figure 3.7 Impact of load and node density in RWP scenario

In this section, we analyze the results when we enable both HDM and SORT for RWP scenario and change a) frequency of message generation and b) node density. Figure 3.7(a) plots results for 50 node RWP runs when video bursts are generated every 37 seconds, 75 seconds, 150 seconds, 300 seconds and 600 seconds while keeping all other values same as in Table 3.5. Figure 3.7(b) plots MOS values for simulation runs while varying node density from 25 to 400 for the same map area. Figure 3.7(c) and (d) plot the costs regarding overhead and CBO.

At a high frequency of SVC burst generation, decoded quality is low, overheads are high, and CBO is low. Overheads are high because most of the content created at source is deleted either on the source or in its vicinity. Source generates new bursts even before the prior burst's content can be shared, thus causing buffers to become full at the source itself. Note that the source has ten times buffer space more than other nodes; still, it suffers from buffer overflow for very high frequencies (37 seconds and 75 seconds) as the gap between contacts with other nodes is higher than the burst gap. For 150 seconds and beyond, overheads drop as more bundles reach the destination (and hence higher MOS scores) while CBO continues to increase since more messages reach the destination and messages in transit stay in the buffer for longer intervals.

Figure 3.7(b) and (d) for change in node counts have a similar trend as Figure 3.7(a) and (c). When a few nodes are present, buffers at source or nodes in the vicinity of source quickly run out of buffer space. Hence, they delete older bundles to accept newer bundles, leading to relatively high overheads (order of 15-18 for node counts of 25[1] and 50). For low node density, buffer space and relay opportunities are limited. Hence, we observe low values for CBO. Overheads taper off as we reach 100 nodes or beyond since maximum copy count used for base-layer is 32. CBO increases slightly while delivery rates increase significantly for 400 nodes, taking the MOS scores above 4.5.

3.4.4. *Analysis of components in SORT*

In this sub-section, we analyze the impact of Operating point control, Replica control, TTL control, Re-distribution and choice of estimation. HDM is disabled in these runs (except for ReDist, where the variance is more when HDM is enabled). To analyze the impact of load and delay targets, in addition to the RWP setting captured in Table 3.5, two additional variants of RWP are used in these runs. *RWP-Fast* generates twice the base

[1] For RWP with 25 nodes, two of the simulation runs (out of 50) did not have any successfully delivery at the destination. The two runs have been skipped while computing overheads and CBO.

frequency (at 150 seconds interval), targets delay of 1 hour and TTL of 2 hours. *RWP-Slow* generates messages at half the speed (at 600 seconds) while it also doubles up the TTL and delay-target when compared to the base (to 8 hours and 4 hours respectively).

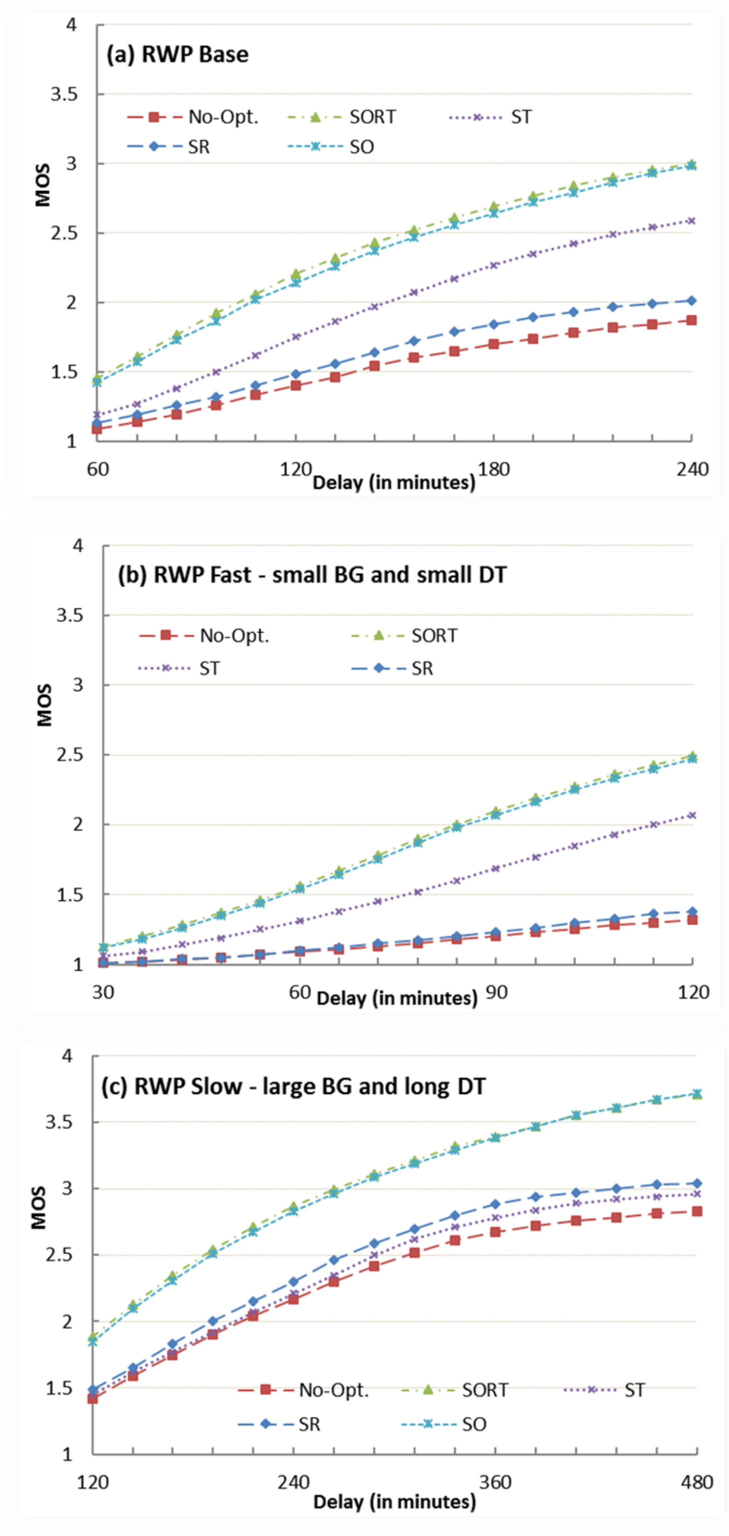


Figure 3.8 MOS impact from SORT components across different load and delay in RWP

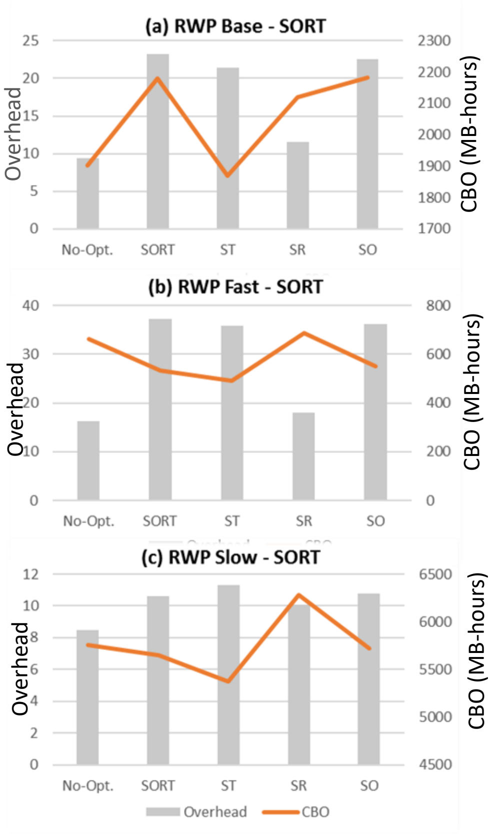


Figure 3.9 Communication and storage costs for SORT components

As shown in Figure 3.8, in all variations of RWP, for quality measured as MOS, operating point adaptation provides maximum improvement. When all three components (ORT) are enabled, it provides slightly better performance than the operating point alone. TTL control provides moderate improvements for Base and Fast scenario, while Replica adaptation provides small improvements for Base and Slow scenarios. Algo. 3.3 applies Replica control only to two highest layers. Hence, its isolated impact is smallest amongst the three.

Figure 3.9 shows that replication control(R) provides maximum savings on overheads for Base and Fast scenarios. However, CBO values are relatively high for R since it keeps higher layers in the buffer (and HDM is disabled on the nodes). ST provides maximum savings on CBO by reducing TTL values when it detects congestion/delay.

Similar to MOS figures, when all three (ORT) are enabled, both overhead and CBO values are almost same as the values for operating point (O). Note that similar trends were observed with SFT and WDM runs. We have included variants of RWP alone, for analysis so that we compare similar mobility patterns.

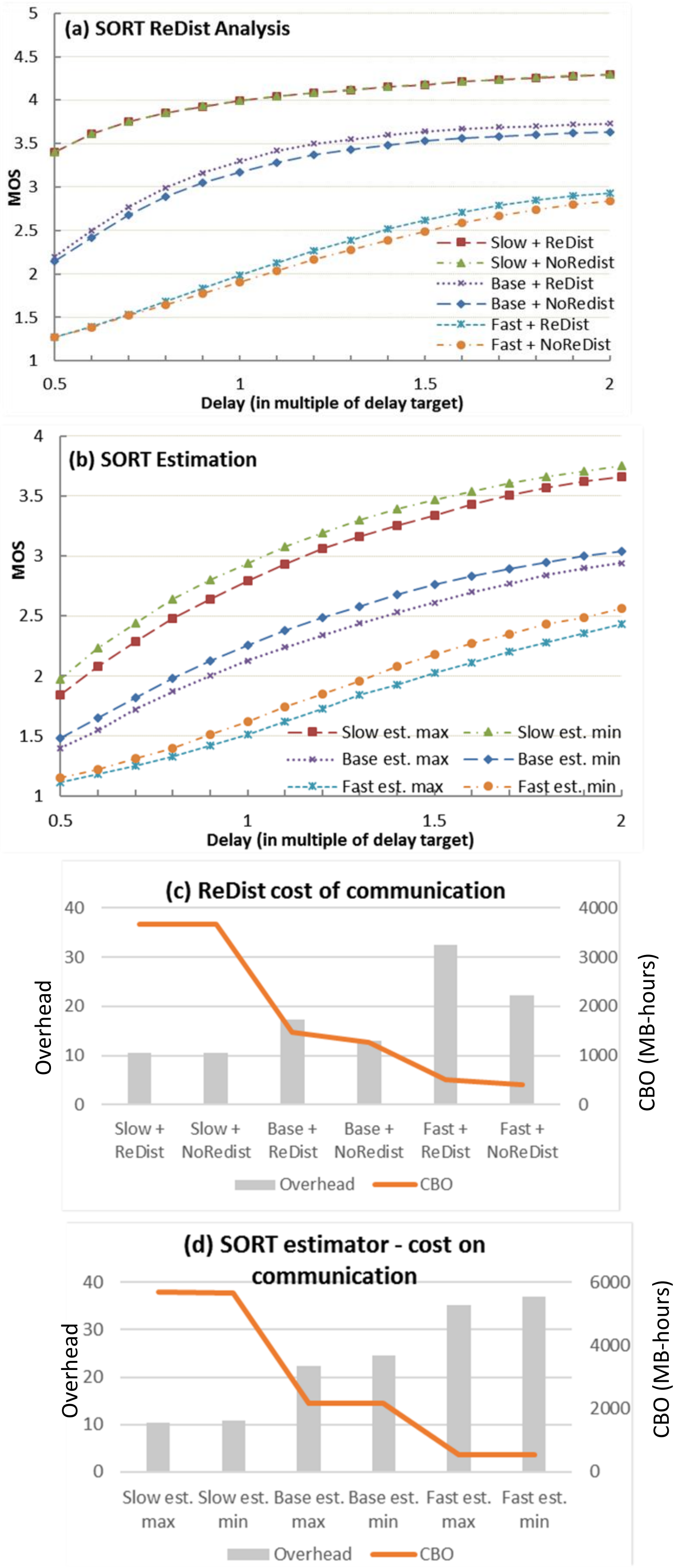


Figure 3.10 Impact from ReDist and Estimation for RWP scenarios

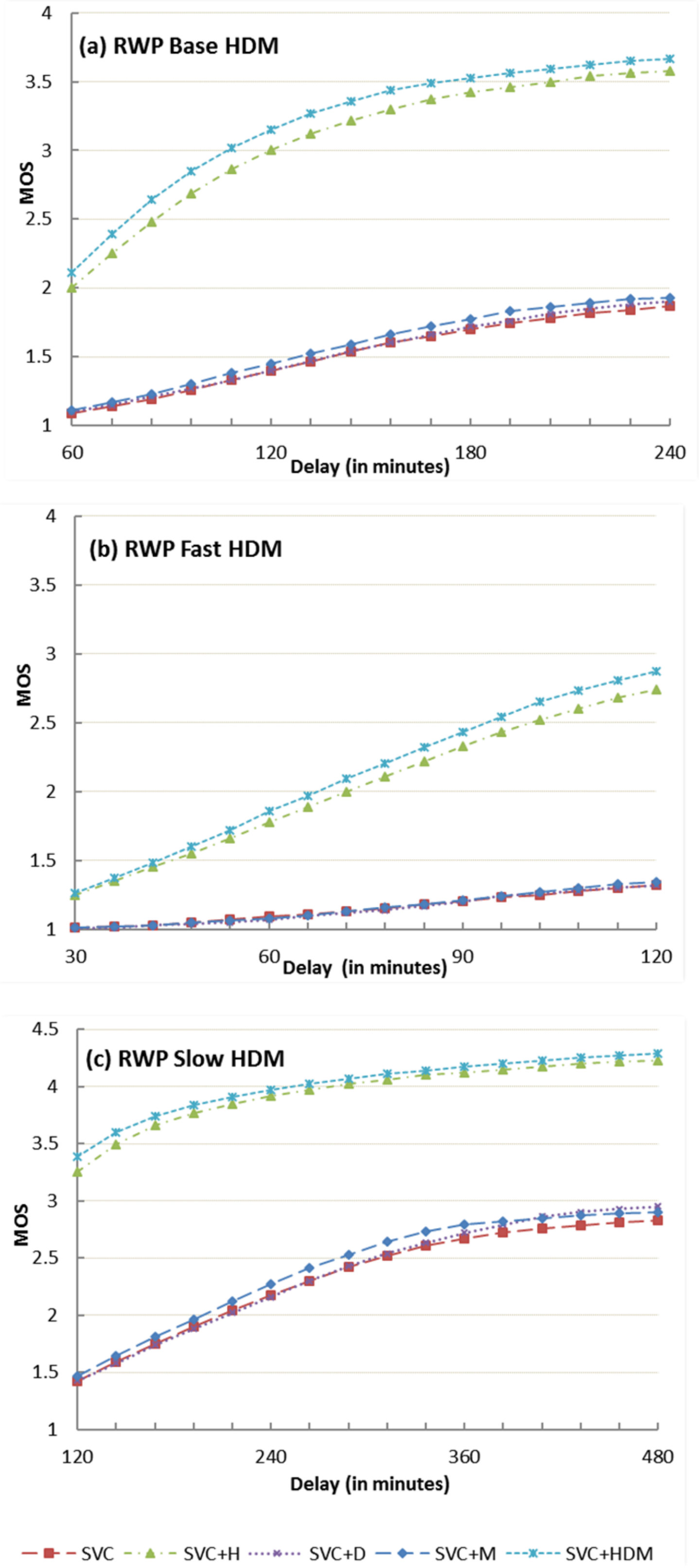


Figure 3.11 MOS impact from HDM components for different load and delay in RWP

Figure 3.10 analyses the impact of ReDist estimation features of SORT. As expected, redistributing copy counts to lower layers impacts the results only if delivery rates are low.

In the Slow scenario, MOS scores are shown in Figure 3.10(a) stay the same irrespective of whether we redistribute or not. On the other hand, for Base and Fast scenario, ReDist increases MOS while also increasing CBO marginally. The overheads for ReDist are very high under the Fast scenario. An in-depth analysis shows that RWP-Slow operates in State-A most of the time, while RWP-Fast computes the State as B and C most of the time. Algo. 3.3 in State-A, does not remove the higher layers; hence there is very little scope to redistribute. On the other hand, in State B and C, adaptation removes higher layers and reduces copy count as well as TTL for some of the enhancement layers. This increases the ReDist to lower layers. In RWP-Fast, bundles are deleted closer to the source, and they do not increase the CBO, but overheads are high because of lower delivery rate.

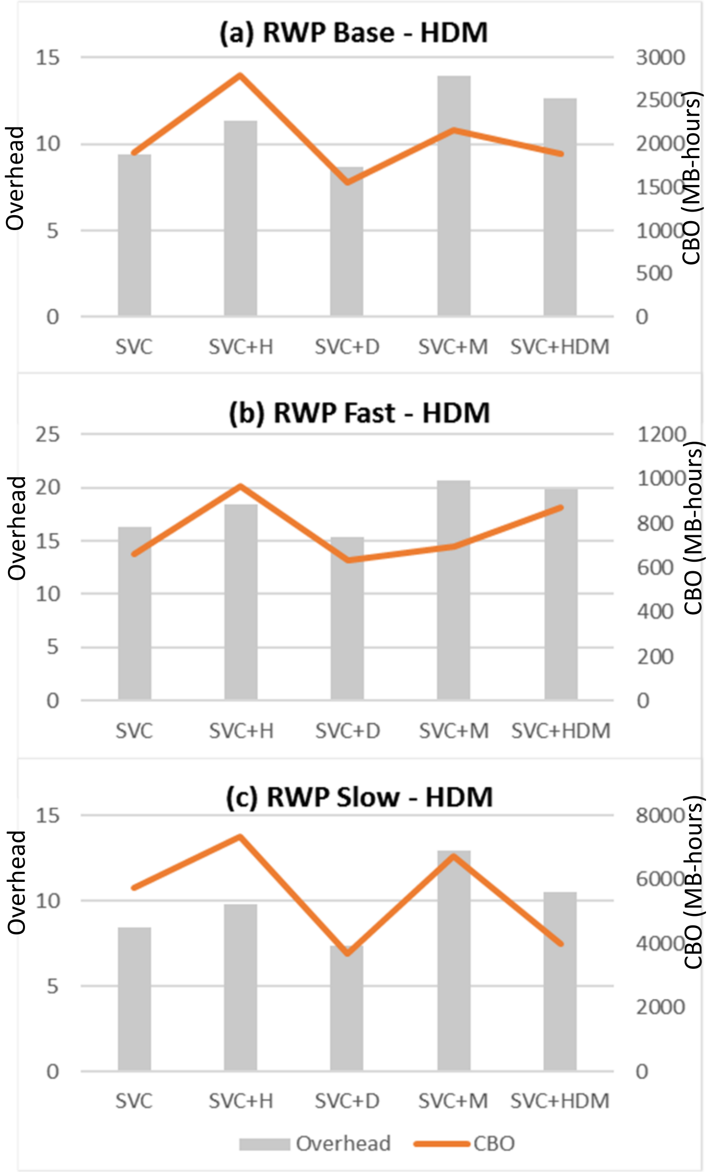


Figure 3.12 Effect of HDM components to communication and storage

When experimenting with choice of estimator, while HDM is disabled, the delivered quality was poor, and all the three estimates (min/max/avg) were very close. Figure 3.10(b) and (d) plots results for the estimate as min/max with HDM enabled on the network. The plot of avg. was in between these values. Hence, we removed it for ease of visualization. Note that minimum estimator pulls the adaptation towards State-A, creating more overheads and increasing the MOS scores.

3.4.5. *Analysis of components in HDM*

In this sub-section, we identify improvements from the three enhancements in SNW - the preference of higher copy count (H), deleting delivered messages (D) and media aware network extension (M). To better identify the contribution from HDM components, we disabled SORT for these experiments.

Figure 3.11 shows that maximum improvements are because of H (preference of higher copy count). As discussed before, this is because lower layers have higher copy counts and hence enabling H helps relay the lower layers faster. MANE provides a slight improvement in Base and Slow scenarios, while D (delete) does not provide any noticeable gain in MOS score. However, when all three (HDM) are combined, there is a slight improvement in quality across the three simulation runs.

Deleting by itself is not expected to improve the MOS score as it comes into play only after bundles are delivered. Analysis of logs showed that the delivery rate increased when delete feature was enabled, but it delivered bundles for multiple layers, and hence the destination did not see any appreciable increase in consecutive bundles received for different bursts. Note that improvements from MANE are lower than that observed in [40]. This is because we have mapped a DTN bundle to each SVC layer for the burst. This mapping is much coarser (order of hundreds of KB) than the mapping to NALU, which are hundreds of times smaller. When MANE triggers such large DTN bundles, the creation of additional copies causes deletion of some of the other SVC bursts, thus offsetting the improvement under heavy load.

Observations on the cost aspects show that “delete” provides a significant saving on both overheads and CBO. Overheads reduce since delivered bundles are not relayed further. CBO improves much more significantly as bundles are deleted as soon as an acknowledgment is received, and additional copies are not created. Figure 3.12(a, c) show that CBO value also benefits from Delete if all three components are enabled, and network load is moderate. Preference for higher copy count increases the CBO and overhead for all scenarios. MANE also increases the communication costs for Slow and Base scenarios, but for the Fast scenario, MANE does not have a cost impact, as most bundles are deleted because of heavy load from the source, and MANE is not able to find a local copy of missing layers.

3.4.6. *Summary of analysis:*

In our experiments, operating point is the most prominent control for application-level adaptation and preference for higher copy count is the most beneficial extension on the network nodes. Note that different choice for OP_ratio, RR_ratio, and MxTtlDrop can make other components of SORT dominant. Since RR_ratio is applied only to two highest layers being transmitted (Algorithm 3.3 – line 7), it does not impact too much on the improvements. Choice of 0.2 for MxTtlDrop in our experiments is also on similar lines. In fact, increasing this value can also make its contribution more prominent. We chose to focus on operating point in our experiments primarily to ensure that the encoder has less work to do when we are not transmitting higher layers.

On-network optimization for MANE provides benefits under low to moderate load while delete feature is an effective way to control the cost of communication. Alternate experiments (not reported in this thesis), where we increased the contribution from MANE showed that it did not scale for high load on the network. Though MANE helped deliver the missing layers, the additional copies were causing further congestion on the network.

Observations across scenarios showed that relying only on routing optimization or only on end host adaptation would not provide the best results. While RWP performs best for on-network optimization, WDM showed better performance for end hosts adaptation, using SORT. Using both SORT and network optimizations always gave the best results.

3.5. Summary

We have proposed SORT, a novel end-to-end adaptive system to improve the quality of streamed video over delay tolerant networks, using SVC. Performance of the system has been analyzed using three different mobility scenarios. SORT provides significant performance gains when transmitting scalable video across different scenarios. MOS scores increased by 0.5 to 0.8 across different scenarios. We also experimented with simple extensions to SNW routing and analyzed their impact on video quality over time. With SVC layers mapped to DTN bundles, when the source assigns higher copy count to lower layers, simple network optimization to prefer higher copy count during bundle relay, provides excellent improvements. In RWP scenarios, it increased MOS scores by 0.7 to 1.5, with low load scenarios demonstrating maximum improvements.

Even though we have used real-world traces (in SFT), we have simulated load-and-network communication for our experiments. There is bound to be some difference between simulation and real-world deployments. Hence, we deployed SVC streaming on an opportunistic network testbed within our campus. The implementation and deployment details are covered in 7.

An area to explore in future is the application of our approach to information-centric networks, especially those involving multiple subscribers (multicast), as well as multiple media flows. Future extension of SORT can use machine learning or similar approaches at the source to estimate bandwidth, delay-target, and TTL to get the best quality. SORT did not try to optimize adaptation for specific scenarios. Instead, we used a simple estimator-based approach without changing the algorithm constants for different scenarios. Other scenario-specific optimizations could also be attempted. For example, in WDM, it is expected that usage of 24 hours earlier data can provide better results, rather than simple EWMA.

SORT only considered changes to end-applications for optimization. Feedback from network elements (other than the destination) for adaptation can provide further optimization. MDC based video communications, especially with single-copy DTN routing (e.g., using custody transfer), is another possible area to explore for scenarios that involve auxiliary nodes.

4. Video Information Extraction

Summary — Most of the contents in video capture from static cameras consist of fixed background. Invariably the important information in such video is associated with some motion. Background Subtraction models help extract such information. Adaptive approaches of background subtraction (like Mixture of Gaussian) do not mandate prior input for object detection, but adaptive approaches suffer from misclassification during the learning phase. We extend Mixture of Gaussian to improve upon the shortcomings of adaptation phase. Video chunks are processed in both forward and reverse directions. Different approaches to combine the output values in both directions are analyzed. Best results are observed when different kernel sizes are used in morphological operations for forward and reverse directions, depending on the temporal offset. Experimental results for standard video data sets and farm-like scenarios demonstrate improvement in detection across numerous frames.

4.1. Introduction

In the previous chapter, we studied how to communicate video streams using scalable video coding to dynamically adapt the transmission to the resource availability on the Opp-Nets. In this chapter, we study object detection, highlight generation and video compression of in farm- like scenarios. As discussed in the context of Figure 1.1, in 1, large areas of farms are monitored using cameras mounted at strategic locations. These cameras are affordable devices communicating over Wi-Fi or Bluetooth- based link layer protocols. In rural scenarios, the actual network may have significantly long delays, and the source node may not be able to retain all the video chunks unless compressed to extremely low bit rates. Continuous video compression to low bit rates significantly degrades the video quality. If we can identify parts of video that do not have perceptual information, we can skip them from compression. In this chapter, we study a scheme to detect foreground objects and communicate only the information related to foreground.

Foreground detection allows us to discard a significant portion of the video where the information is perceptually static. The challenge is to accurately segment the regions that are to be discarded, from the regions to be retained. It is obvious that for such scenarios, the motion is associated with portion of the video to be retained. Hence we study how to improve segmentation accuracy by using bi-directional motion detection.

Typical captures, for farm monitoring, involve large open areas with trees and other vegetation. The intent is to detect human/animal or inanimate objects intruding into the farm without requiring explicit human interaction. Similar to previous chapter, videos are stored and processed in chunks, varying from tens of seconds to a few minutes, to extract key information. It is only the extracted key information that is sent over the Opp-Nets. Focus of this chapter is to reduce the amount of video data to be compressed, hence reducing the payload at the application layer, without missing out on the information being conveyed. The improvements from other optimizations studied in this thesis will be complementary to what we study in this chapter. E.g., the foreground video frames can still be compressed and transmitted using SORT.

To ensure that the information being conveyed is accurate, and at the same time un-necessary information is not being compressed, we need to improve identification of regions with significant motion while discarding movements generated by the perceptually fixed objects. For example, swaying branches, tree-tops, etc. show lots of small movements across the frames, but they are insignificant information for guard or owner of the farm. We do not expect the nature of foreground objects or background to be known in advance. A priori knowledge-based approach could cause unknown objects to be missed. Though the wider work for complete automation in farm scenario can involve object tracking across frames, path-identification, etc., they are not within the scope of this thesis. We focus only on improving segmentation accuracy so that we save on resource demands of Opp-Nets by compressing smaller regions of frames and skipping frames that do not have any motion. Lower demands should also imply better quality and lower delays for delivered content.

As discussed at the beginning of this thesis, what we study in this chapter can have applications beyond farm like scenarios. E.g., we can apply these improvements on information extraction from archived videos, conserving storage space for archived videos for close circuit television based recordings, reducing human viewing effort for recording from multiple camera feeds, etc.

Next section of this chapter provides an overview of the key object detection techniques and includes an overview of Back-Ground Subtraction (BGS) using Mixture of Gaussian (MOG) model. Subsequently, we provide the implementation details of bi-directional video processing for object detection. After that, we provide the experiment details followed by the results and related analysis. The final section concludes the chapter.

4.2. Background Work

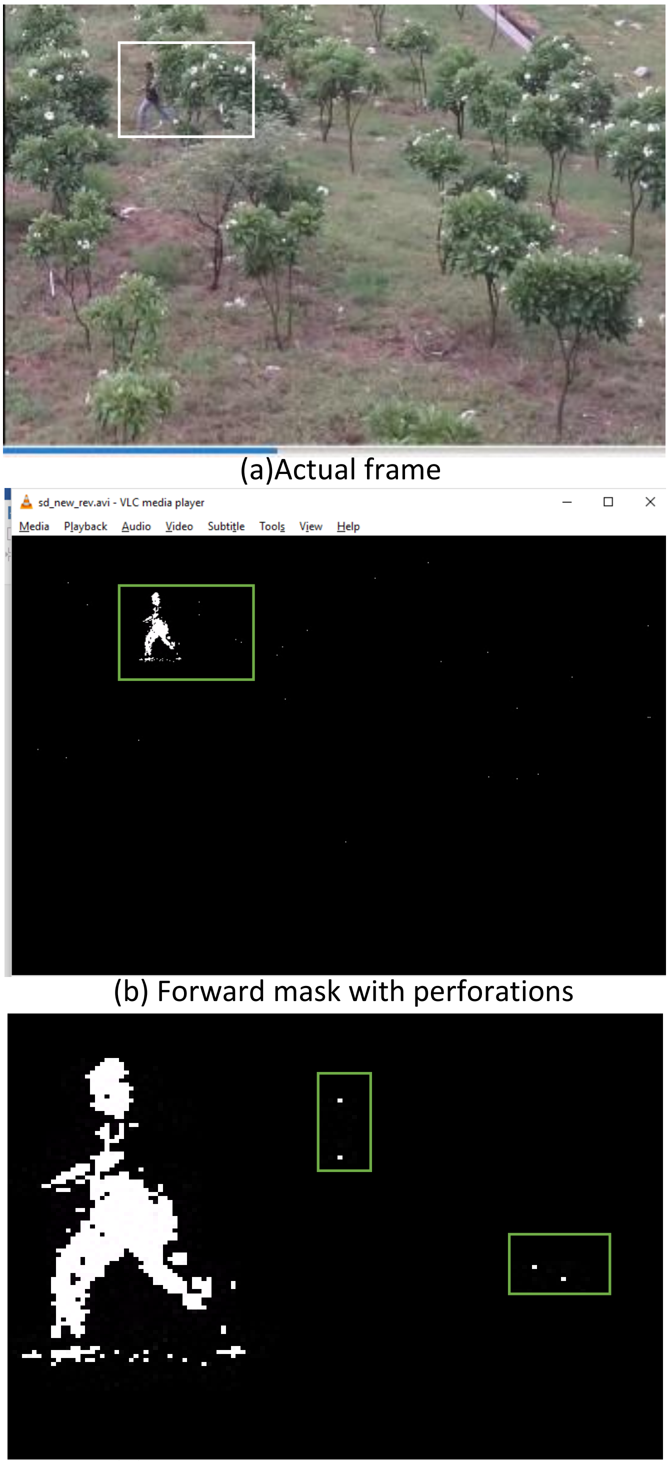
Multiple approaches have been developed to detect motion of objects in the video. Three of the common approaches are a) Optical Flow; b) Background subtraction (BGS); and c) Color histogram-based methods [52]. Histogram-based algorithms need prior information wherein the object to be detected is tagged in one frame and the algorithms continue to track the object in subsequent frames. It attempts to do the best fit for the histogram across the frames. Optical flow based approaches identify key points in a frame (e.g., sharp corners or areas with a sudden change in contours) and track them across the frames. Applying optical flow in farm like-video generates lots of possible points in the video frames, leading to scalability issues. Hence we decided to use background subtraction based scheme for farm-based scenario [53][54]. The background subtraction algorithms perform well with modern systems that have more graphical processing cores dedicated to the algorithm [55].

Motion detection using background subtraction involves segmenting each video frame into background and foreground regions. The background region stays relatively static throughout the length of the video. The foreground is the region of pixels that move across the frames while mostly maintaining its shape and color attributes.

4.2.1. *BGS - Mixture of Gaussian: A brief overview*

Background subtraction methods rely on creating a model for the background. Each frame is compared to the background model to find the difference and identify the regions of motion. Background subtraction using Mixture of Gaussian (referred to as BGS-MOG in subsequent parts) differs from the other background subtraction methods on how the background model is created. The background model is continuously updated for every frame that is processed. Every pixel of the background is represented as a mixture of

Gaussian distributions (mean, variance and weight) [56]. This approach allows efficient handling of gradual illumination changes (e.g., shadows, clouds, etc.) as well as adapts to swaying trees and branches. BGS-MOG does not require bootstrapping (externally identifying the object) or prior learning (training). It does suffer from camouflage issues (especially for slow-moving objects) and does not work if camera view is not static.



(a)Actual frame

(b) Forward mask with perforations

(c) Zoomed in view of the perforations

Figure 4.1 Sample perforated mask from the data set and its zoomed view

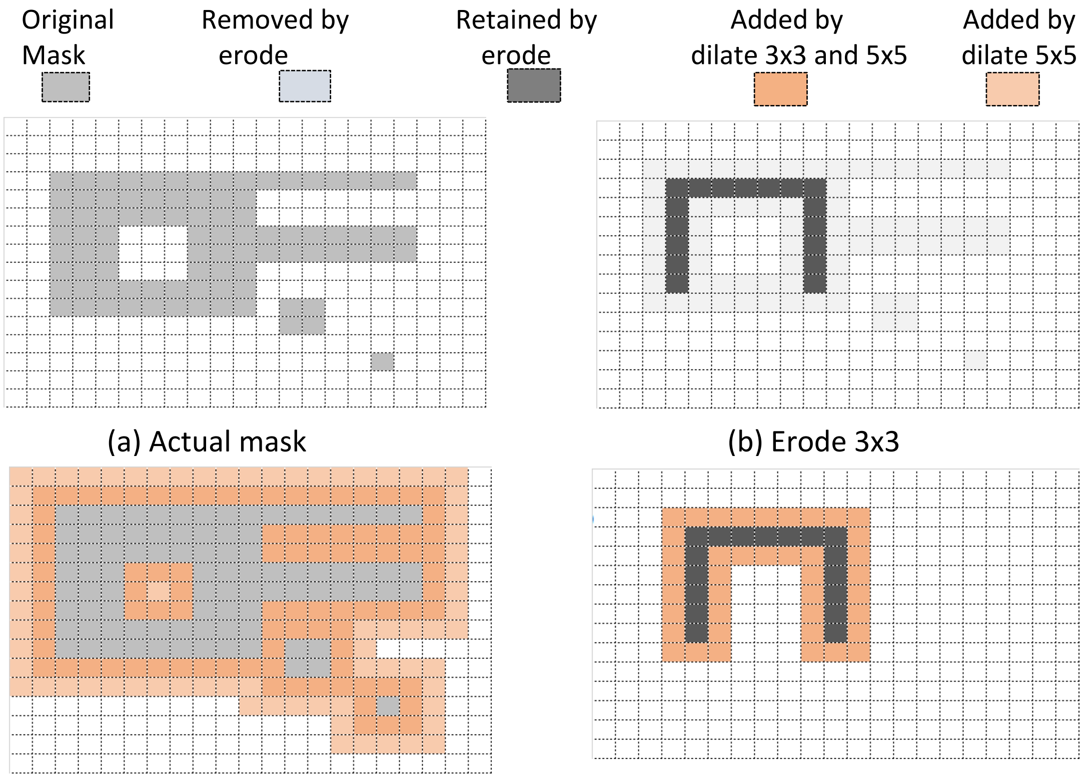
BGS-MOG maintains a mixture of Gaussian distributions for each pixel in the frame. When processing a new frame, the value of each pixel in the new frame is compared to the corresponding distributions in the MOG. If a match is found, the corresponding distribution is updated for mean, variance and the weight. If no match is found, a new distribution is added to the mixture (if number of distributions exceeds the maximum allowed entries, then the least probable distribution is replaced by a new distribution). This new distribution is allocated the default weights and variance. All weights are normalized following updates/additions and the mixture model is sorted in descending order for weights. A learning rate (α) drives the weights applied to updates and additions.

A Gaussian mixture model is a more accurate description of the background as compared to any single distribution. The adaptive nature of the algorithm accounts for lighting changes and works even if no pristine background model was available.

When pixel does not match or matches against a very low weight in the mixture, it is marked as foreground in output mask. The mask provided by BGS MOG for a frame is generated on individual pixels without a sense of adjacency. This results in a perforated mask with varying pixel values for a given object. Figure 4.1 provides the original frame, mask for the same and zoomed in view of the mask for one of our test frames. For ease of visibility, a small part of the frame is zoomed in to show some of the motions from branches and leaves. Such small dots are present all over the frame in most cases.

4.2.2. *Erode and dilate on foreground mask*

Since the foreground is generated as perforated mask, it needs to be cleaned up to remove areas with the wrong detection. Morphological operations are matrix-based mathematical operation on images. These operations alter a pixel based on the values of the pixels in its neighborhood. The choice of the neighborhood (area and shape) affects the results of the operation. These operations are commonly referred to as erode and dilate. The erode operation reduces the thickness (shrinks) around the edges whereas dilate increases the thickness. If random small objects are present, erode tends to remove them while dilate will end up joining them. Size and shape of kernel decide how effective the removal or joining is.



(a) Actual mask

(b) Erode 3x3

(a) Dilate 3x3 and 5x5

(b) Erode 3x3 followed by dilate 3x3

Figure 4.2 Mask and its impact on erode and dilate with square-shaped kernels

Figure 4.2 gives an example of erode and dilate on a binary mask. It can be noticed that erode removes not only the salt and pepper noise, but also removes the pixels from the edges of the shape. Dilation tends to add pixels to the mask, in the neighbourhood of existing values.

4.2.3. *Shortcomings of BGS-MOG*

a) Spurious motion in initial frames:

The foreground mask created for the initial frames show a lot of motion since the background model (MOG) is still under construction. Learning rate (α) drives the weight for the pixel value in the current frame. If moving objects are present in the first frame, the algorithm incorrectly classifies the initial region where the moving object was present, as part of the background. In subsequent frames, a lot of incorrect motion is reported where the moving body was originally present. Inaccurate motion is removed after a few frames, as the background model adjusts itself to the real values for background pixels. Adjustment period depends upon the learning rate α.

b) *Missed interiors of large objects and trailing edges of small objects:*

Trailing edges of some objects are missed if they are narrow along their motion path. This is because mixture model has not adapted to the object. The MOG has a significant weight for the original background and hence when the background is uncovered, it is not flagged as a motion in the mask.

Similarly, for large objects (say a slowly moving yellow van) with a large length along the motion path, though the leading and training edges are identified as motion masks, the interior is not properly identified. This is because the color of bus is added to MOG and after a few frames, its weight becomes significant enough to be classified as background.

4.2.4. *Impact of α on BGS-MOG shortcomings*

It's difficult to ensure that initial frame does not have any motion; hence the initial frames will have false positives. To reduce this learning period, a very high learning rate (α) can be used. However, a high α causes slower moving objects to be mislabelled as background. This makes the tracking of slower moving objects difficult. Generally, a compromise is made on the learning rate and some inaccuracy in reporting motion in the first few frames is accepted. We attempt to reduce this inaccuracy.

4.3. Bidirectional Processing

To mitigate the limitations identified in previous section, we started processing the video in both directions, on the assumption that trailing edges will be better identified in reversed video. Bi-directional processing deals with chunks of video of 30 seconds to a few minutes. The stream is individually analyzed from start to end, and from end to start. Results obtained from the two analyses are then aggregated to generate a mask that has improvements in identifying the region of motion.

Figure 4.3 covers the initial steps of bidirectional processing. The first step (A) in the process is to create a separate reversed video of the given video chunk (last frame to first frame). In step (B), BGS MOG is applied on both video chunks. This results in two separate masks for the video- one is for the first-to-last frame video (henceforth called the forward stream) and the other for the last-to-first frame. The BGS MOG stream for the reversed video frame has the masks in last to first order. The two streams are aligned so that, for a given frame number both streams provide the masks for the same frame in original video. This is done in step (C) by reversing the mask stream for the last to first frames (subsequently referred to as the reversed stream).

The two streams can be used individually to detect motion in a given frame. However, both suffer from the shortcomings mentioned in the previous section. To improve detection accuracy, the two masks from the streams have to be aggregated to improve the accuracy. Three aggregation methods were experimented upon, and for all these methods, dilation and erosion have been used. The three differently sized square kernels used were, SMALL(3x3), MEDIUM(5x5) and LARGE(7x7).

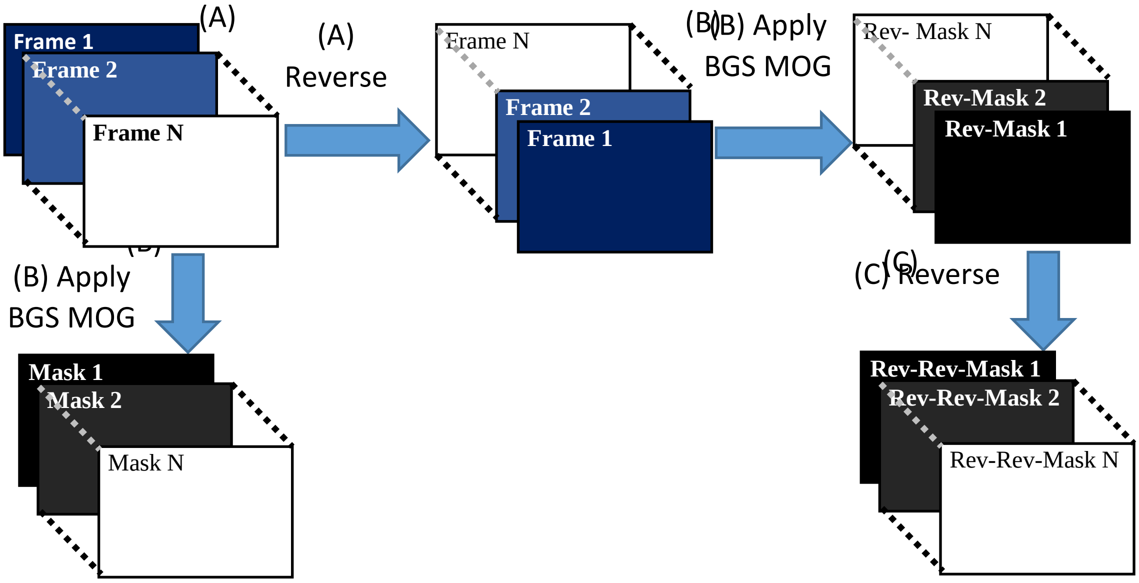


Figure 4.3 Creation of reverse and forward mask streams

Initial aggregation methods used were Bitwise OR and Bitwise AND. Before the Bitwise operation, the forward and reverse stream the frames are dilated using the MEDIUM sized kernel. After the Bitwise operation, the resulting mask is eroded using LARGE kernel. This is done to minimize the addition of noise from the individual streams. This may result in loss of information. Note that integer-based mathematical addition and multiplication were not used since the masks are binary in nature and bitwise operations are faster.

The accuracy of the various aggregate streams was then compared to the accuracy of individual forward and reverse streams. For comparison, the individual forward and reverse streams were dilated and eroded with the similar sized kernel (MEDIUM), to smoothen the edges and to fill in small gaps within the shapes. Figure 4.4 captures the processing done for individual streams (forward, reverse) and aggregated streams (AND, OR).

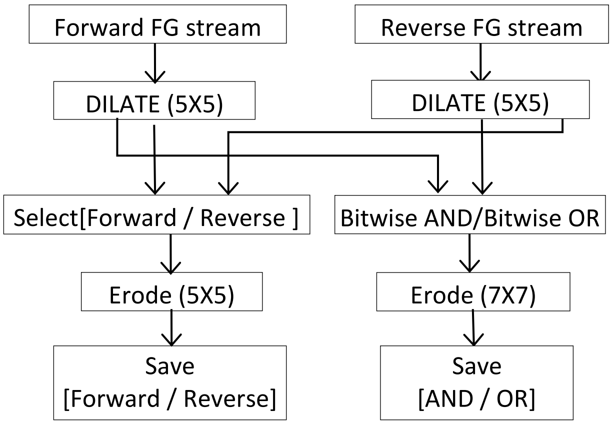


Figure 4.4 Morphological operations for individual and simple aggregated streams

As expected, the Bitwise OR aggregation method raised the noise level in the mask. The effect was more pronounced in the initial and terminal part of the video. Similarly, the Bitwise AND aggregation method leads to loss of information, especially around the edges of objects.

To overcome these issues, we implemented a third aggregation approach based on Weighted Bitwise OR. For this method, the masks from forward stream were preferred during the terminal part of the video, and the masks from the reverse stream were preferred during the initial part of the video. The approach used morphological operations as weight functions for individual streams. Differently sized kernels were used as function weights. Algo 4.1 captures the details of this approach.

For the first 5% of the video, forward stream is in learning phase. Similarly, for the last 5% of the video, reverse stream is in learning phase. The FG stream, not in learning phase, is given more weight by using a LARGE kernel for dilation. The other FG mask (corresponding less preferred stream) is dilated with the SMALL kernel. After the Bitwise OR operation, the resulting mask is eroded using a LAGE kernel. During the middle portion of the video (5%-95%), both the forward and the reverse streams are given equal preference and dilated by the MEDIUM kernel.

Algorithm 4.1 Weighted Bitwise OR

```
Initialization
    a. SMALL kernel = matrix of size (3x3) with all 1;
    b. MEDIUM kernel = matrix of size (5x5) with all 1;
    c. LARGE kernel = matrix of size (7x7) with all 1;
Acquire forward_MOG_stream;
Acquire reverse_MOG_stream;
Initialize output_stream;
for all frames in video_stream do
  acquire forward_mask from forward_MOG_stream;
  acquire reverse_mask from reverse_MOG_stream
  if frame in (0 to 5% of video_stream) then
    mask1 =Dilate(forward_mask, SMALL kernel);
    mask2 =Dilate(reverse_mask, LARGE kernel);
  else if frame in (95% to 100% of video_stream) then
    mask1 =Dilate(forward_mask, LARGE kernel);
    mask2 =Dilate(reverse_mask, SMALL kernel);
  else
    mask1 =Dilate(forward_mask, MEDIUM kernel);
    mask2 =Dilate(reverse_mask, MEDIUM kernel);
  end if
  combined_mask = BitwiseOr(mask1, mask2);
  final_mask = Erode(combined_mask, LARGE kernel);
  output_stream.add(final_mask);
done for
```

In the first 5%, for the reverse MOG stream, the model is mostly accurate; hence it has a better representation of actual motion. A Bitwise OR of the two dilated masks creates an aggregate that has dilated objects from both the streams, i.e., highly expanded regions from the reverse MOG and minimally expanded regions from the forward MOG stream. Subsequent erosion of the aggregate mask using the LARGE kernel diminishes the regions of motion that are only contributed by the forward MOG (as a smaller kernel dilated it). Most of the small random regions, contributed by forward MOG, are removed and larger regions, contributed by forward MOG, are diminished in area. For regions contributed by the reverse MOG stream, the factor of dilation is the same as the factor of erosion; hence no detail is lost for them.

Similar logic is used for frames in last 5% of the video, which ensures that motions identified in forward stream are prominent. For the remaining part (i.e., in the middle), both the streams are dilated by the same factor. A kernel of MEDIUM dimensions is used on both the forward and reverse MOG streams.

4.4. Experimental Setup

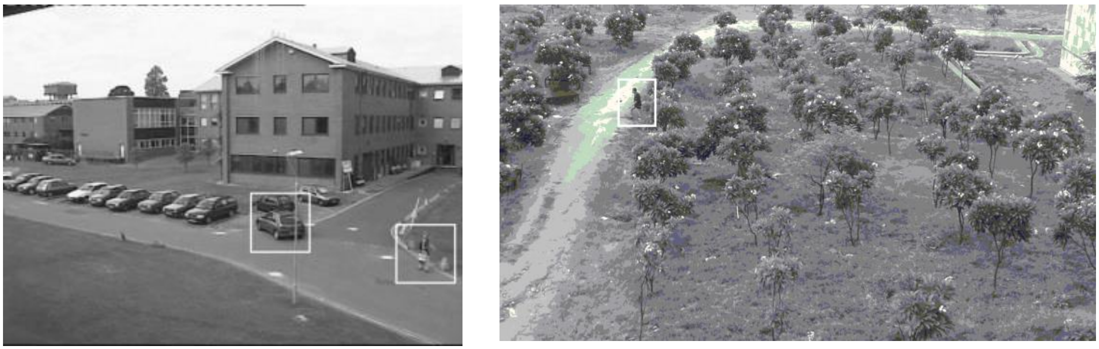
In the experimental runs, Background Subtraction Model with Mixture of Gaussian is used with a maximum of 20 mixture channels, and learning rate (α) is set as 0.005. For the morphological operations, square kernels were used with dimensions as given in Algo 4.1.

The algorithm is executed using the existing implementation of BGS-MOG "Improved adaptive Gaussian mixture model for background subtraction" that was used for all the experiments [57]. Open-CV 2.4.9 running on Ubuntu 14.04 was used for these tests.

4.4.1. *Test Videos*

Two video sources are used.

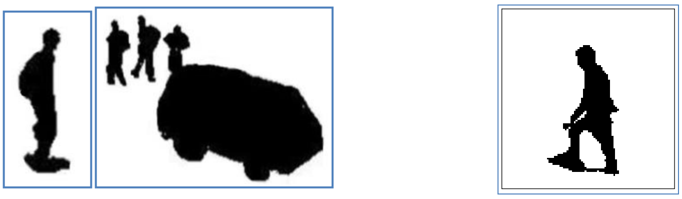
- Video from within the authors' campus is chosen such that it presented challenges that were similar to the farm-monitoring scenario. (Here-in-after, referred to as "internal video"). The video was shot in HD Ready resolution at 30 fps. Due to the large distance, the commonly moving objects are seen relatively small. The paths are such that objects are occluded by tree and shrubs; the trees and shrubs having moderate swaying motion.
- Clip from PETS 2001 video -Camera 1; from here on referred to as the PETS video [Reference: PETS2001 datasets (University of Reading, UK)]. It has motion of people and vehicles at 25 fps.



a. PETS - frame 400

b. Internal - frame 400

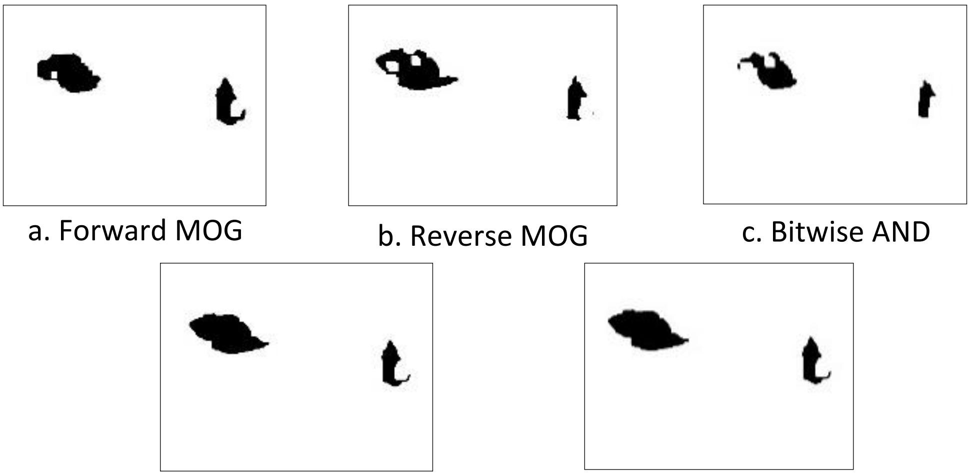
Figure 4.5 Sample Video frames



a. PETS - Video

b. Internal - Video

Figure 4.6 Ground truth video frame 400



a. Forward MOG

b. Reverse MOG

c. Bitwise AND

d. Bitwise OR

e. Weighted OR

Figure 4.7 Resulting foreground masks for frame 400 zoomed 4x to bottom right

Across the video sources, series of representative frames were chosen. Each frame series consists of four individual frames (e.g. frame numbers 5, 10, 15, 20). Frame series are chosen from the start, middle and end of the video. The actual region of motion (Ground Truth) for the representative frames has been manually identified.

For each video source, chunks of 30 seconds are used in experimental run. Sample reference frames are captured in Figure 4.6. The ground truth for these sample frames is shown in Figure 4.7. Similar work was repeated for all the twenty-four frames (3 series of four frames each, for both video sources).

For each video, the metrics are calculated for forward BGS MOG, reversed BGS MOG, Bitwise OR, Bitwise AND, and Weighted OR. The representative masks for frame 400 of PETS video are shown in Figure 4.7 (a. through e.).

4.4.2. *Metrics For evaluation*

[58] identifies three metrics for the accuracy of motion detection. 1) Missing Foreground, 2) Added Foreground and 3) Rate of Misclassification [58].

Missing foreground refers to the percentage of foreground pixels mislabelled as background pixel. It’s calculated for each representative frame as

$$\mathrm{MF}_{rel} = \frac{\text{Missing foreground pixels in the mask}}{\text{Number of foreground pixels in the Ground truth}}. \tag{Eq. 4.1}$$

Added foreground is a measure to determine the degree to which background pixels are mislabelled as foreground pixels.

$$\mathrm{AF}_{rel} = \frac{\text{Added foreground pixels in the mask}}{\text{Number of background pixels in the Ground truth}}. \tag{Eq. 4.2}$$

MFrel and AFrel do not account for the degree of misclassification for a missed foreground pixel or added background pixel. A falsely detected pixel that is closer to the actual foreground is better, compared to a farther pixel. This measure is provided by the metric Rate of Misclassification. It’s given as:

$$Fsc(n) = \frac{1}{F(n)} \sum_{j=1}^{F(n)} \left(\frac{d_{err}^{i}}{D} \right). \tag{Eq. 4.3}$$

Where F(n) contains set of pixels in MF ∪ AF. d^{j}_{err} is the distance of the jth misclassified pixel from the closest contour of the ground truth. D is the length of the frame diagonal.

4.5. Results and Observations

Table 4.1 captures the values for AF_{rel}, MF_{rel} and F_{SCrel}, across the three frame series for both video sets.

Table 4.1 Results (MF_{rel} and AF_{rel} multiplied by 1000)

			Forward	**Reverse**	**Bitwise OR**	**Bitwise AND**	**Weighted OR**
Internal Video	**Series 10**	MF_{rel}	0	0	186	757	36
		AF_{rel}	17.56	0.25	0.14	0	0.07
		F_{SCrel}	298.02	1.98	11.47	1.11	1.65
	Series 400	MF_{rel}	3	15	265	480	265
		AF_{rel}	1.58	1.13	0.23	0.04	0.23
		F_{SCrel}	7.9	8.62	8.4	9.33	8.4
	Series 800	MF_{rel}	152	152	411	508	197
		AF_{rel}	1.02	3.66	0.41	0.07	0.43
		F_{SCrel}	9.63	86.66	21.75	3.02	17.42
PETS Video	**Series 10**	MF_{rel}	140	3	45	207	6
		AF_{rel}	1.62	1.16	2.64	0.36	1.60
		F_{SCrel}	29.77	3.25	51.60	2.81	13.52
	Series 400	MF_{rel}	36	138	36	271	36
		AF_{rel}	5.44	5.95	5.85	2.83	5.85
		F_{SCrel}	165.25	179.13	179.37	166.45	179.37
	Series 700	MF_{rel}	197	156	24	449	24
		AF_{rel}	10.78	13.08	13.76	4.48	13.76
		F_{SCrel}	269.28	289.20	294.46	260.27	294.46

MF_{rel}: Bitwise AND is the worst performer as it missed the foreground objects consistently. This is visible in Figure 4.7 (c) where a large part of the car is missed.

AF_{rel}: Bitwise AND, shows only those regions that are common to both the forward and reverse MOG streams. Since most noise generated by the two streams does not overlap, a very low AF_{rel} value is obtained. In the initial frame series, reverse streams have the lowest AF_{rel} followed closely by Weighted OR approach. Correspondingly, for the end frame series (of the PETS video) the reverse stream has the lowest values followed very closely by Weighted OR.

F_{SCrel}: It measures the distance of the misclassified pixel (missing foreground and added background) from the nearest contour. Bitwise AND has the best result in most cases closely followed by the remaining methods. The reason for good values of F_{SCrel} for Bitwise AND is due to the missing foreground pixels being very close to the contours in out examples. Since the contours are small, this effect is not prominent.

We conclude from Table 4.1 that

- Forward and reverse MOG streams do not perform well in the initial and the final regions of the respective videos. Otherwise, they have good performance (low AF_{rel} and F_{SCrel}) and moderate MF_{rel}.
- Combination approaches (both AND and OR) use moderate kernel for dilate and erode, hence they have lower MF_{rel} during the middle part of video. OR is significantly better than AND for MF_{rel}.
- Near start/end of video chunks, either of forward or reverse are best, while Weighted-OR is a close second (sometimes even tying for best performance).

When comparing the three aggregation approaches, Weighted OR algorithm effectively removed the noise while performing well for missed foreground across all parts of the video.

4.6. Summary

Efficient motion detection techniques enable the creation of accurate tracking solutions. The proposed method provides a novel way to improve motion detection. The improvement comes at the cost of additional processing required to reverse the video stream (step A and C of Figure 4.3) and analysis of the reversed video as well as the original video. For internal video dataset, the compressed output is shared at www.swifiic.in/sambv/index.html. Fourteen megabytes were compressed to below 600 kilobytes for similar perceptual quality.

Experiments with bidirectional video processing showed excellent results for Weighted OR. It improved segmentation efficiency while keeping false positives within the bounds. Future work can study the effects of different learning rates, kernel shapes and kernel sizes across different video datasets. Further, bidirectional approaches can be applied to other object detection and tracking algorithms that do not expect static camera view (E.g. [59][60]).

Such advances can be put to use in the fields of surveillance, video compression, event generation, etc. The method is not real time but is implemented in close to real time using chunks of video. For an Intel i7 (3770 @3.4 GHz), without external GPU support, 30 seconds of video is processed in approximately 90 seconds. It is expected that using solid-state disks and external GPU; this algorithm can process chunks of videos in near live manner.

5. IMPROVING FAIRNESS AND MANAGEMENT OF OPP-NETS

Summary — Some of the nodes in sparse areas of opportunistic networks may not have a good quality of service compared to other nodes. It is important to identify such nodes and prioritize their content. If each node attempts to identify such nodes in a distributed manner, it adds to computation, storage and communication complexity for each node. The partial view of nodes will also be sub-optimal and extremely difficult to track and audit. In this chapter, we study the usage of a central node that collects global view of Opp-Net performance. This centralized view can help ensure good quality of service, while also simplifying management and monitoring. It can improve DTN routing with minimal increase in routing complexity on individual nodes. We study central node assisted routing where message forwarding and relay continues in pure distributed manner. Global view computed by central node augments the routing decision on individual nodes to improve fairness for poorly served nodes. In the absence of "global view", Opp-Net routing continues in a distributed manner without the benefits of fairness improvement.

Additionally, we study the extra overhead and congestion effects when all application messages are relayed through central node (say for policy enforcement, malware detection, management, etc.). In this case, the central node becomes a key part of routing as all messages are made to pass through it. Other than increasing the manageability of Opp-Net traffic, it is expected that this node may provide other value additions like billing, transcoding, etc. as required. We analyze the delay and congestion impact for such a scenario.

5.1. Introduction

As discussed in 1, delay and disruption tolerant networks (DTN) [4] come into play when end-to-end communication using mesh or ad-hoc networks is not feasible. Node mobility is the primary enabler for DTN as it emphasizes on the store-carry-forward behavior of the bundles (payload or messages for DTN).

As we cover in 7, we have experimented on DTN based platform in rural scenarios with a focus on applications for intra-community communications. This is a scenario where

DTN based terrestrial communication service is provided on a long-term basis for a village community. In such deployments, people with low mobility or nodes in the periphery of the network are treated unfairly. It is observed that they have lower delivery ratios and significantly higher delays.

Revenue generation is essential to make DTN rural deployment sustainable. Any revenue model mandates feature like, accounting, billing, security, compliance with local laws and quality of service. In this context, challenges in pure distributed DTNs are a) lack of metering and accounting of resource usage, b) tracking and enforcement of quality of service for individual nodes, c) enforcement of policies to comply with local laws, and d) ability to do security scans (like IDS/IPS in commercial TCP/IP based networks).

To mitigate the challenges of purely distributed DTN, we explored deploying a central device to simplify user management, billing, policy enforcement, etc. The owner/operator of the central device will be able to transparently bill the end users for the services they have received. Central device, referred to as CoHub (centrally optimizing hub) in this chapter, creates a global view of nodes in the Opp-Net. Such a global view on CoHub can assist the routing protocols on individual nodes to target specific quality of service for end nodes.

One of the applications for this rural platform (and primary focus of this thesis) is sharing highlights from farm monitoring videos. Rather than sending complete video, only masked highlights as discussed in 4 may be compressed and sent. Since video generates large content (megabytes) on a periodic basis (30 seconds to a few minutes), it is important to know the quality of DTN service, so that the highlight creation and compression can be adapted to the network conditions. Information shared by CoHub can help drive such decisions.

This chapter proposes a new routing extension “Quality of Service Enforcing Centrally Optimized Routing”, referred to as QoseCo in rest of the thesis. We use Spray-and-Wait [14] as the base protocol, and demonstrate QoseCo extension on top of it. Spray-and-Wait (SNW) is chosen because it can control the overheads using the number of replicas (L) as discussed in 2. Performance of QoseCo + SNW is compared with vanilla SNW. Message forwarding is done by all nodes based on information shared by the CoHub. In the absence of CoHub, QoseCo will degrade to vanilla SNW. Other protocols like PRoPHET can also be adapted with QoseCo (for example by adapting difference in utility values for poorly served nodes and delete policies when buffer gets filled-up).

An extension of QoseCo is explored where all application messages are forced to be relayed via the CoHub. This is referred to as QoseCo with Central-Relay [QoseCo-CR]. Relaying via central nodes is expected to cause congestion and make CoHub the single point of failure, and still for commercial deployments, it may be mandatory for security scanning and policy enforcements. Usage of multiple CoHub (horizontal scaling) or vertical scaling (using multiple antennae's of faster link layer technologies) is some mitigation approaches for avoiding congestion and single point failures. Such mitigation techniques are not part of this thesis. We attempt to measure only the impact on delivery ratios for QoseCo-CR for a single deployment of CoHub.

Structure of the remaining part of this chapter is as follows: Section 5.2 provides a brief overview of related DTN routing protocols and quality of service related work. Section 5.3 provides overview of QoseCo and QoseCo-CR along with algorithmic details. Section 5.4 has simulation setup, and results & analysis. Section 5.5 concludes this chapter.

5.2. Related Work

As discussed in 2, Massri and others in [10] have identified three major components in DTN routing– 1) queue management, 2) forwarding decision and 3) replication. Routing decisions in such cases can use the infrastructure-based approach with auxiliary nodes like mules and throw-boxes or use participant nodes (IMN) for store-carry-forward. In this chapter, we use a hybrid approach wherein "infrastructure" is used to improve routing decisions, but the actual routing is done using participant nodes in a distributed manner.

Some of the newer protocols explore human / social aspects of the person carrying the device to drive the routing decisions. This increases the computation complexity on the nodes as each node needs to have visibility beyond its own contacts to calculate social characteristics like centrality and between-ness. For DTN networks with power-law, [61] explores DTN routing via hub nodes, where (multiple) hub nodes have higher connectivity and offload computation load from other nodes. Another set of routing algorithms uses an oracle or deterministic approach to compute future contact probabilities and drive routing decisions accordingly. But when these protocols are applied to Opp-nets, especially for rural people who need not keep a strict schedule, they perform poorly.

When adding an extension to an existing routing protocol, the base protocol should be as simple as possible. Naïve Spray-and-Wait [14] was the obvious choice as it allows good control of overheads, besides being simple regarding computation and storage complexity on all nodes. To target fairness, QoseCo allows higher replica count (L) to be used for messages with poorly served nodes as origin or destination. During forwarding, QoseCo allows higher share of replicas to be given to well-connected nodes. Similar to SNW, PRoPHET can also be considered. PRoPHET computes the likelihood of delivery to destination using prior node encounters. In PRoPHET, a message is forwarded to the other node, only if its likelihood of delivery is higher than the node having the message by a predetermined threshold. Changing the threshold value determines how many replicas will be created on the network. This threshold can be relaxed for poorly served nodes if PRoPHET is to be adapted for fairness.

5.2.1. *Quality of Service*

Hay and Giaccone [62] have explored QoS for DTN routing when contacts between nodes are known in advance or can be predicted in advance (e.g., Bus routes etc.). Such knowledge is used to model the DTN network as a time-dependent graph for optimizing QoS. Sandulescu [37] uses special messages under nominal load from the source node to destination node and collects feedback over time. After that, source adapts the transmission to observed network behavior. QoseCo targets doing away with such needs, by periodically updating about poorly served nodes to CoHub, which shares it to the whole network.

5.2.2. *Fairness in DTN:*

Pujol et al. [63] attempt to measure fairness in the share of load on individual nodes. Fan [64] et al. measure fairness in delivery ratios for increasing number of contacts. During relay of messages, their work attempts to locally compute the utility value of each message, with delivery farness of messages as a goal. Since each node tries to compute fairness without a global perspective, the impact may vary, based on a node-location and deployment scenario.

Similar to [64], we attempt to improve fairness in delivery ratios. To the best of our knowledge, control of delivery fairness w.r.t. end nodes had not been dealt with in any earlier work. QoseCo uses historical contact and delivery information to adjust the priority of nodes. Since complete graph need not be built on each node, QoseCo scales quite well, though it has a time lag in sharing the global view.

5.3. QoseCo Routing Extensions

As discussed, we study two sets of routing extensions. The first extension only attempts to improve fairness. The second extension also attempts to send all bundles via the central node (CoHub).

QoseCo: All nodes periodically share their state information with CoHub. Based on the information received, CoHub generates a consolidated view and updates all nodes (broadcast) with control information. This communication adds to the transmission overhead (below 2% for our experiments), but it helps move the computation to CoHub. CoHub periodically distributes the global context to all nodes. In the absence of information from CoHub, the node degrades itself to naïve routing protocol (SNW in this case).

QoseCo-CR: If complete control of the deployment (e.g., lawful interception, virus scan, etc.) is required, all traffic is relayed through the CoHub. This allows the CoHub to also include value-added features like billing, adaptation of content, non-repudiation, etc. as discussed in 7.

Major decision points for the two extensions are captured below.

5.3.1. *New message from the origin node*

Replication count (L) for the message is set by the origin node, based on the last routing control message received from the CoHub. For poorly served nodes, higher counts are allowed. Algo 5.1 captures the message generation as source node. Defualt_Max_Replica is the default have of L. Higher values are allocated for poorly served node as covered in step 6 of the algorithm.

Algorithm 5.1 New Message on originating node (simplified)

```
1.  function newMessage(Message M, Host origin, Host dest)
2.     L = Default_Max_Replica;
3.     if (dest or origin in poorMap)
4.        lowQual = MIN ( getQuality(dest), getQuality(origin) );
5.        L = L * (1+AverageQuality / lowQual);
6.     if(routing == QoseCo-CR)
7.        m.setMetadata("actualDest", dest);
8.        m.destination = CoHub;
9.     m.setMaxReplica(L);
10.    m.setDestination(destination);
11. end
```

For QoseCo-CR, the destination is set as CoHub, and the actual destination is added as metadata to the message. Once CoHub gets the message, it does the required processing and relays it to the actual destination.

5.3.2. *Feedback to the CoHub*

All nodes periodically send status updates to CoHub (management payload). The update message includes, a) information about node contacts (contact count and total duration) with other nodes; b) count of messages generated and delivered from/to end applications on the node; and c) list of messages received.

5.3.3. *Consolidation of data (Global-View) at CoHub*

When a CoHub receives status update messages from the nodes, it adjusts the metrics for the nodes. Periodically the CoHub multicasts routing updates to all nodes. This multicast status includes, a) list of poorly served nodes and their priority, b) list of nodes that are good carriers with their priority, and c) list of recently received messages by destination nodes. Poorly served nodes are identified as nodes that have reported very few contacts and/or whose packets have not been delivered (either as origin or destination). Good carriers are nodes that have large number of contacts with different nodes. Exponential weighted moving average is used to reduce impact of old contacts.

When nodes get multicast routing updates from CoHub, they delete all messages in their local cache that are successfully delivered to the destination nodes. They cache details of good carriers and poorly served nodes for subsequent routing decisions and new message creation.

5.3.4. *Routing decision: Message forwarding between nodes*

Algorithm 5.2 Forward Message Between Nodes (simplified)

```
function startTxfr(Msg[] pendingMsgs, Host[] connList)
    if (CoHub in connList && CoHub in getDestinations(pendingMsgs))
        candidateMsgs = filterOnDest(pendingMsgs, CoHub);
        sortOnOriginNodeQuality(candidateMsgs);
        transferAndDelete(candidateMsgs[0]);
    else
        trgtMsgs = filterOnDest(pendingMsgs, connectedList);
    if(trgtMsgs!= EMPTY)
        sortOnOriginOrDestQuality(trgtMsgs);
        transferAndDelete(trgtMsgs[0]);
    else
       txrList = sortOnEndNodesAndLforRelay(pendingMsgs, connList);
       relayAndAdjustL(txrList[0].msg, txrList[0].remote);
end
```

During contacts, nodes exchange messages with each other according to Algo 5.2. The message transfers are attempted as follows: A) Deliver messages with CoHub as destination, if one of the connected nodes is a CoHub; B) Deliver messages with destination as connected node; C) Or relay other messages and adjust copy counts.

If there is a tie w.r.t. message or connected host, the tie is broken as follows: a) messages moving towards CoHub, b) messages with nodes as source or destination of poor quality, c) current replica values of messages, and d) relay to host which are good carriers. The function relayAndAdjustL gives a higher number of copies to nodes that are in “Good Carriers” list.

5.3.5. *Message deletion when buffers are full*

Expired messages are deleted first. Further deletion of messages happens, as required, based on when the message is received weighted by priority of source and destination nodes.

5.4. Simulation and Results

QoseCo routing and forwarding is implemented in ONE [41]. In this section, we first present how metrics for the experiments are computed. We then present the simulation scenario. Thereafter, we present the results for the various experiment. For Inter-event Gap (Congestion), results are discussed only for delivery ratio and fairness. Overhead is skipped, since overheads shoot up significantly as delivery ratio drops. All metrics are average values for eight randomized runs.

5.4.1. *Metrics computation and impact of buffer space*

A. **Delivery Ratio**: This is the ratio between the number of bundles received at the destination to the number of bundles transmitted from the source.

$$delivery\_ratio = \frac{(total\ bundles\ delivered)}{total\ bundles\ created} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Eq. 5.1}$$

B. **Delivery Delay**: This is the average time interval from the time of payload creation at the source to the time of delivery at the destination.

C. **Overhead Ratio**: This metric denotes the cost of delivery. It is computed as per the equation is given below:

$$overhead = \frac{(total\ bundles\ relayed\ or\ created\ - total\ bundles\ delivered)}{total\ bundles\ delivered} \quad . . . . \text{Eq. 5.2}$$

The performance on Opp-Nets is primarily constrained by a) buffer space on nodes; b) communication bandwidth, that is the contact duration and network speed for sharing the information during contact; c) energy capability of portable devices and d) computation capability of nodes. Further, the scenario-specific contact pattern, size and frequency of application payload, and time-to-live (TTL) for payload also impact the performance.

To improve the delivery ratio frequently, routing algorithms may attempt to increase the replica count or the source may increase the TTL values. Such approaches provide improvements if the Opp-Net is not resource-constrained or congested. Once any of the resources get saturated, the nodes will notice increased wastage. Sometimes, despite higher overheads, the delivery ratio may drop, and delivery delay may increase.

For example, allowing for more replicas even when buffer space is full will cause drops for older messages (assuming simple FIFO for queue management). In such cases, a large payload, when allocated a higher replica count, end up triggering a drop for multiple earlier messages, some of which could have been delivered to the destination, had they not been pre-empted from the queue. Based on Eq. 5.1 and 5.2, it may be observed that while delivery ratio dropped, overheads will be further amplified for this example.

5.4.2. *New metric for QoseCo*

Reporting functions were added to ONE, to appropriately capture details of management packets and include measure of fairness metric. Jain's throughput fairness index [65] is used. These metrics are captured in Table 5.1.

Table 5.1 Computation of Delivery Ratio, Overhead, and Fairness

Parameter	Value
K	Count of nodes that participate either as source or destination in communication attempts.
T_Delivered	Total messages successfully received at destination
T_Started	Total messages sent
T_Relayed	Count of successful forwards of messages between all nodes
N_Success	Number of messages successfully delivered for a specific node (either as source or destination)
N_Attempt	Number of messages for a node (either as source or destination)

$$\text{Delivery\_Ratio} = \text{T\_Delivered} \,/\, \text{T\_Started} \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Eq. 5.3}$$

$$\text{Overhead} = (\text{T\_Relayed} - \text{T\_Delivered}) \,/\, \text{T\_Delivered}. \ldots\ldots\ldots\ldots \text{Eq. 5.4}$$

$$\text{N\_Xfactor} = (\text{N\_Success}/\text{N\_Attempt}) \,/\, \text{DeliveryRatio}. \ldots\ldots\ldots\ldots \text{Eq. 5.5}$$

$$\text{Fairness} = (\sum \text{N\_Xfactor})^2/(\text{K} * \sum (\text{N\_Xfactor2})) \ldots\ldots\ldots\ldots \text{Eq. 5.6}$$

5.4.3. *Simulation Configuration*

Two mobility models being used - (a) Working Day Movement model – WDM[42]; and (b) Helsinki city scenario [41] using shortest-path-based movement (also referred to as Random-Way-Point RWP). Generic details about these scenarios are captured in section 3.3.3.

To target rural scenarios, 64 nodes are used, and simulations were run for longer durations. Longer duration was chosen, as compared to [41] because rural areas of developing countries will not have Internet connectivity at home/office. For RWP, 37 pedestrians, 3 people with cars or two-wheelers, and 21 static nodes were used along with three trams. For WDM, numbers from [42] were scaled down by a factor of sixteen to get 64 end-user nodes. For both the scenarios, static CoHub node was added. CoHub location was manually chosen near road junctions to ensure good contact opportunities.

Table 5.2 Default Settings for Simulation Runs

Parameter	Value	Parameter	Value
Duration	4 days	NumCopies	64
Event Gap	8 minutes (randomized)	EWMA Alpha (QoseCo)	0.002
Message Size	750 KB (randomized)	NumPoorService (QoseCo)	16 nodes
Message TTL	8 hours	NumGoodCarriers(QoseCo)	8 nodes

QoseCo update messages are pretty small (typically 4-16 KB for CoHub routing updates and 1-4 KB for node status updates). They have not been included in further analysis since they are smaller by order of hundred times. For a sample run, 200 megabytes of traffic relay was caused because of QoseCo management load while application payload generated 13 gigabytes of traffic.

Note that the router implementation extends the existing SNW implementation in ONE to add support for Hub and exchange of messages between hub and other nodes.

5.4.4. *Experiments for moderate load over varying period*

Execution was done for simulation periods of 0.5 days, 1 day, 2 days, 4 days, 8 days, and 16 days. The results are plotted in Figure 5.1. As simulations run for longer interval, delivery ratios improve in all scenarios. This is because higher percentages of messages are received within the simulation time. QoseCo follows similar trajectory to SNW with a slight degradation. The degradation is observed because management messages compete with application messages during contact opportunities. QoseCo-CR has lower delivery ratios. This is because routing via CoHub requires more hops and it also increases congestion in the vicinity of the CoHub.

As duration increases, fairness improves since messages are generated with random nodes as source and destination. Both QoseCo and QoseCo-CR show significant improvements till 2-4 days, with slope tapering off for 8 days and 16 days.

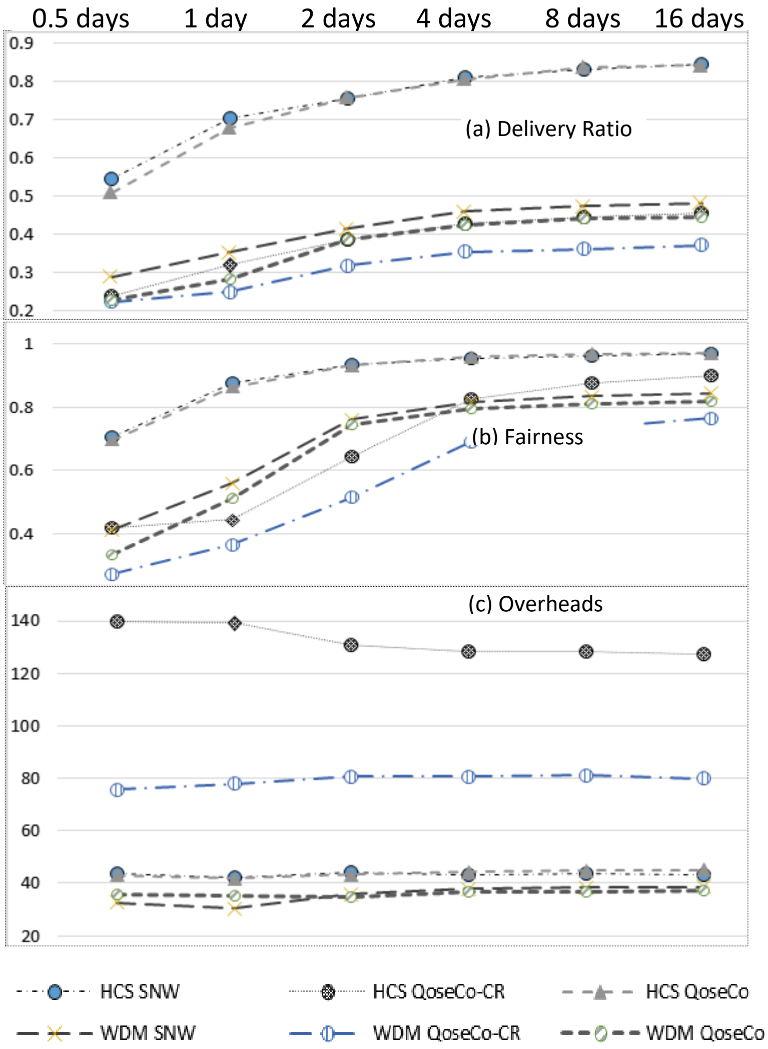


Figure 5.1 Simulation Duration Results

For runs of less than 2 days, lots of messages are not yet delivered; hence overheads are higher. Overheads are higher for QoseCo-CR as it first sends messages to CoHub and then CoHub relays to destination. Moreover, RWP has lot more contacts; hence it creates many more copies. Overhead results for QoseCo were marginally higher as it allowed a larger number of replicas for poorly served nodes. All through QoseCo had fairness pretty close to SNW, despite poorer delivery ratio for QoseCo, on account of added management messages.

5.4.5. *Inter-event Gap- Congestion*

Events were randomly generated with an average gap of 30 seconds, 1 minute, 2 minutes, 4 minutes, 8 minutes and 16 minutes. This scenario allows congestion to happen in cases where messages are generated at two minutes and below. Figure 5.2 plots result for fairness and delivery ratio for four-day run.

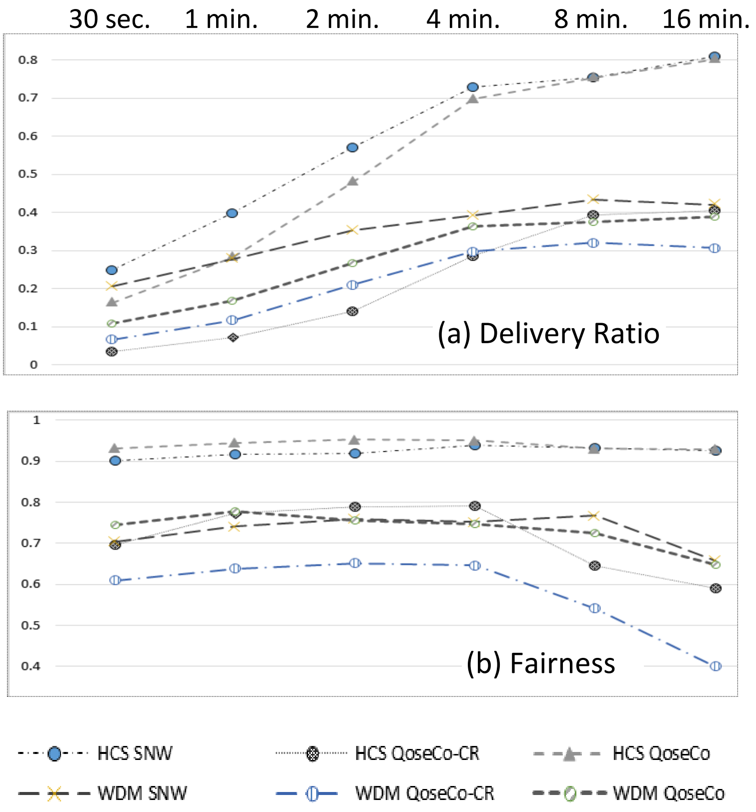


Figure 5.2 Varying gaps between events

When events are generated twice every minute, almost all nodes participate in communication, and most of them observe high losses, though the nodes on the periphery will be worse off for WDM. For HCS, QoseCo does well to improve fairness in this scenario as it has values around 94% while SNW is around 90%. A similar trend is also observed for WDM (70% vs. 74% for 30 seconds). At lower frequencies since less number of nodes participating in communication, slight degradation is observed in fairness for all cases.

QoseCo-CR observes significantly higher packet loss when message frequency is high (low delivery ratio). This is because the messages get relayed via the CoHub, and for all messages that get relayed, a new message is created.

5.5. Key Observations and Analysis

HCS scenario is completely random concerning mobility paths and destinations, while WDM is somewhat synchronized. With choice of 64 nodes in simulation, and messages randomly generated between nodes, it was observed that QoseCo improves fairness by around 4% as load increased (towards the left in Figure 5.2).

Improvements for fairness in QoseCo are more pronounced when network is congested (frequently generated messages). This happens despite the overheads of QoseCo update messages. Poorly serviced nodes are tracked centrally. By moving this computation to central nodes, computation load for other nodes was reduced. The trade-off was the addition of QoseCo update traffic which was less than two percent for most scenarios.

QoseCo-CR is frequently worse off by a factor of about two. For 30 seconds in HCS, it degraded beyond a factor of two when compared to QoseCo as severe congestion set in the vicinity of the CoHub.

5.6. Summary

We studied DTN routing extensions, which used SNW as underlying routing algorithm to demonstrate centrally assisted fairness optimization to DTN routing. Using simulations, we also showed how centralized routing extension helps improve Quality of Service under congestion. We demonstrated the impact of QoseCo for different mobility scenarios, to fairness in delivery ratio, as the quality metric.

We also studied overheads when all messages are relayed through a central node. The cost of relaying messages via central nodes is within 2X for most scenarios but increases significantly as congestion sets in. These observations imply that for low load scenarios we can deploy a single CoHub (as experimented using SWiFiIC in subsection 7.2), but for higher load we need to have alternate designs to avoid congestion and SPOF. That is the reason why Vectors (in sub-section 7.3) does not redirect traffic via a central node.

Possible extensions to this study include use of more than one location for CoHub(s) to reduce congestion and avoid single-point-of-failure. Moreover, QoseCo does not explore location and time relationships for the contact. While we allowed additional copies to improve fairness, one can also extend this by reducing the number of copies for nodes with high delivery ratio (e.g., above 95%). Only delivery ratio is considered for fairness; delay in delivery can be another metric for optimization.

6. ENERGY EFFICIENT DISCOVERY IN OPP-NETS

Summary—Neighbor discovery in sparse opportunistic networks can require significant energy. Generally, discovery occurs by sending and waiting for probe messages and responses respectively from nearby nodes. Algorithms dynamically vary intervals between probes to conserve power. Based on analysis of the recent discovery approaches, we study an adaptive discovery algorithm "speed and time-based energy efficient probing (STEEP)", which uses details of latest 'connection up' event and node speed.

We also study the impact on discovery when nodes turn off the radio interface to conserve power, which may typically cause higher discovery failures. Extensive experiments conducted using real-world traces and working day model show that STEEP provides 30% - 50% power savings for discovery in delay tolerant networks (DTN). It also demonstrates good results for DTN routing as well as better adaptation to density changes.

6.1. Introduction

In the earlier chapters, we explored optimizations at application and routing layers for improving video delivery over opportunistic networks for videos captured in a farm like scenarios. In this chapter we focus on saving energy on intermediate nodes, to stretch their active life time. The video capture source and video destination are primarily with end users and can have the ability to replenish their batteries. For example, external battery with a solar panel can provide charge for the capture device. But the intermediate nodes that help carry the video chunks over Opp-Nets need to ensure that they do not drain off their energy while the nodes are mobile.

As discussed in 1 And 2, it's the aspect of "carry" helps complete the Opp-Nets over time. Before a node can exchange bundles, it needs to discover if there are nodes in its vicinity for communication. Generally, the discovery process uses probe requests (or scans/page requests based on the wireless technology used). However, in a strict sense, depending on the wireless protocols, there is a subtle difference between scanning and probing. In this work, we use the term "probe" for all such activities. A probe is a trigger signal from nodes to discover their neighbors. Active nodes, within a neighborhood, must respond to the probe to complete the discovery process.

In traditional networks or for static nodes, neighbor detection can be a continuous mechanism. Special network infrastructure elements such as routers, gateways, base stations or access points, provide such features. For example, in access point (AP)-based Wi-Fi networks, mobile nodes discover the AP, either by waiting for beacons from it or by sending probe messages to it. Nodes that want to connect to the access point can initiate an association request after receiving the beacon. If the mobile node does not want to wait for beacons in each channel, it can send a probe request. The access point will respond to successfully received probes, and nodes can proceed with association thereafter. In both scenarios (probe or beacon), the AP needs to spend energy continuously for discovery. For other wireless protocols, mobile nodes may use a slightly different mechanism to complete the discovery process, but they will invariably expect the infrastructure support to carry out continuous discovery (e.g., ZigBee controller and cellular base-stations). Power management is a challenge in high-density cellular networks. Certain schemes [66] optimize the spectrum and power allocation in the licensed band. Since the base station has a centralized view of all the nodes, the base station can allocate the resources optimally when used as the association controller [66].

In contrast, most of the opportunistic network nodes lack support from such infrastructure elements. A node should send a probe request in the hope that another node may be nearby. Portable and mobile devices with limited energy resources cannot perform continuous probing, as it will drain the battery drastically. The energy limitations are paramount, for deployments away from power sources or in disaster-like scenarios. While energy harvesting and better technologies that store power will continue to provide larger power capabilities to nodes, efficient discovery algorithms can stretch the active time of nodes in delay tolerant networks.

To conserve energy, nodes tend to deploy different probing strategies. By performing periodic probing (as opposed to continuous probing), a node may miss some short contacts or may discover contacts after a delay. Periodic probing approaches involve a trade-off between the number of probes and the number of missed/delayed contacts. Efficient neighbor discovery algorithms should reduce the number of probes with minimal discovery failure and a minimal reduction in contact duration. Furthermore, an efficient neighbor discovery algorithm should not hurt DTN routing.

Other than spending energy for communication, nodes also consume energy in activities such as computations, data storage, user interfacing, and controlling / polling sensors. In this work, we restrict ourselves to communication-related power consumption for neighbor discovery.

We make the following contributions in this chapter:

- We design an adaptive discovery mechanism (STEEP), where each node independently controls inter-probe time based on speed and latest contact to conserve probe energy and improve discovery rates.
- We evaluate the impact of different discovery algorithms on DTN routing while varying the replication overheads.
- We analyze the performances of the discovery algorithms when nodes attempt to sleep for different targets. We also analyze the scalability of algorithms by varying node densities. To our knowledge, no prior work has conducted a scalability analysis of discovery in opportunistic networks.

STEEP combines the knowledge of the current speed and the time since the last significant event (either connection up or a change in node speed), to compute the interval between probes. Since most of the contacts occur when nodes move, STEEP can outperform the prior discovery algorithms. If a node is traveling at a slow speed or if the time since the last significant event is large, STEEP increases the gap between probes, thus conserving energy.

For non-infrastructure-based deployments (e.g., sensors with limited power), algorithms such as STEEP are expected to help increase the overall deployment lifetime of DTN-based communications. Even for nodes that can harvest energy, STEEP provides additional energy savings, allowing nodes to stay active for longer intervals.

The remainder of this chapter is organized as follows. In next section (6.2), we present an overview of the existing device discovery algorithms while thoroughly exploring eDiscovery, PISTONv2, and STAR. We provide the details of our proposed algorithm STEEP in 6.3. Section 6.4 discusses the energy consumption of wireless interfaces and analyzes the performance of STEEP. In Section 6.5, we present the simulation conditions. The simulation results and relative performances are discussed in Section 6.6. Finally, Section 6.7 summarizes the findings of the study and describes possible extensions.

6.2. Related work

Pozza and others [67] discuss the taxonomy and classification of various neighbor discovery algorithms for opportunistic networks. At a high level, they categorize discovery algorithms into mobility-agnostic and mobility-aware discovery. We can further sub-classify mobility-agnostic algorithms as either synchronous or asynchronous. Synchronous-mobility-agnostic algorithms work under a controlled and homogeneous setup but do not adapt well in heterogeneous environments. Asynchronous-mobility-agnostic approaches generally rely on alternate low power channels or temporal overlap inferred from prior probabilistic knowledge.

Choi et al. [68] propose a mobility-agnostic asynchronous sleep-scheduling algorithm called adaptive cyclic difference set (ACDS), where the nodes undergo periodic sleep modes during idle inquiry times based on precomputed schedules. The ACDS attempts to send probes during the statistical overlap of active cycles across nodes. Shih et al. [69] proposed an additional low-power radio receiver for discovery. After discovery, the node uses the primary receiver for high-speed communications. Since mobility-agnostic approaches rely on either synchronization, overlapping schedules or the presence of alternate low-power channels, such approaches are too restrictive for dynamic and heterogeneous environments.

Complexities in mobility-agnostic approaches for dynamic and heterogeneous environments include the following:

- The absence of global synchronization capabilities across different types of nodes.
- Failure to guarantee overlapping active slots as the number of nodes changes dynamically.
- Complex mathematical computation and scheduling when nodes have different features, such as different radio ranges or different numbers of channels/antennae.

A mobility-aware approach uses either the temporal knowledge, spatial knowledge or both. Temporal knowledge includes network properties such as the node arrival rate and contact duration, while spatial knowledge includes the speed, geographical locations, etc. With the collected data, each node independently predicts the network behavior and adapts the discovery process. These approaches vary the interval between the probes (inter-probe time or IPT), to conserve energy.

The authors of [70] discussed discovery using the short-term arrival rate (STAR). As a mobility-aware temporal approach, STAR uses a heuristic algorithm for arrival rate estimation and then adapts the probing interval based on the estimates. PISTON [71] (a mobility-aware spatial approach) computes the IPT based on the maximum node speed and recent discovery count. Subsequently, the developers of PISTON extended the algorithm to continuously adapt the IPT based on the instant node speed. This extension is referred to as PISTONv2 [72]. Bo Han et al. developed eDiscovery [73], which adapts the inquiry window and inquiry interval based on recent encounters in Bluetooth-based deployments. We describe eDiscovery, PISTONv2, and STAR in detail in this section. For details on other approaches, readers can refer to [67].

6.2.1. *eDiscovery*

In eDiscovery[73], the IPT is adjusted based on the number of nodes (peers) detected by current and prior probes. If the current probe and the prior probe find no peers, the IPT is incremented by “inc_NP + r”. If the prior probe makes no discoveries but the current probe is successful, the IPT follows “base_I + r”. If both the current probe and prior probe detect some peers and their values differ, then the IPT is either incremented or decremented by "I", based on whether the current probe detects more or fewer nodes. Algo 6.1 presents a simplified version of eDiscovery. For our work, we assume probe-triggered discovery. Thus, the scan duration is insignificant and is not included in the algorithm. The authors of [73] set inc_NP and base_I to 10 seconds, while a value of 1 second was used for I. The value of r (random factor) was 0 for 80% of the invocations and -1 second or +1 second for every 10% of the invocations.

Algorithm 6.1 eDiscovery probe and IPT computation

```
if CurrentTime > (LastChecked + ComputedIPT) then
    peers = Probe for Neighbors and get count of discovered peers;
    If (peers == 0 and last_peers == 0) then
        ComputedIPT = ComputedIPT + inc_NP + r;
    else if (peers > 0 and last_peers == 0) then
        ComputedIPT = base_I + r;
    else if (peers > last_peers)
        ComputedIPT = ComputedIPT – I;
    else if (peers < last_peers)
        ComputedIPT = ComputedIPT + I;
    end if
    ComputedIPT = ensureIPTinLimits(ComputedIPT);
    last_peers = peers;
    LastChecked = CurrentTime;
end if
```

eDiscovery offers a simple calculation for IPT without the need for excessive historical context and does not require complex inputs, such as location detection or speed estimation. It also adapts to density changes around itself, using discovered node counts for current and prior probes. Since eDiscovery uses only the knowledge of contacts in the present and prior probe, it is likely to miss short contacts under the low-density scenario. Additionally, eDiscovery misses some of the bursts of contacts that are commonly associated with human mobility. For example, early in the day with almost no contact during the night, the IPT value is expected to be high. Thus, as a person travels to work, eDiscovery is likely to miss the initial contacts. Moreover, a node that continues to have the same set of contacts is expected to maintain the same IPT. Consequently, such a node sometimes continues to probe at a high frequency even in the absence of new nodes in its vicinity.

6.2.2. *PISTONv2*

Because new contacts will occur only when the nodes move, PISTON [71] computes the next IPT value based on the node speed. It computes the IPT using the max speed (B) and radio range (R), according to Eq. 6.1. Subsequently, for PISTONv2 [72], the authors revised the process such that the IPT is modified after every probe based on the instantaneous node speed as opposed to the maximum speed, which was the basis in their earlier work. Since all new contacts occur because of mobility and the IPT is adapted after each probe, PISTONv2 achieves much better discovery than PISTON. In PISTONv2, α is set to 1 to achieve the best performance. It is expected that in real deployments, α will be determined externally based on the trade-off between power conservation and discovery targets.

$$IPT = R/(2*\alpha*B) \quad \text{Eq. 6.1}$$

For PISTONv2, the number of probes sent by each node does not vary even if the number of nodes in the vicinity changes. As the node density increases, the energy spent by the node for discovery remains nearly the same. While this condition is beneficial when the node density is high, for a low-density scenario, the number of probes does not decline, thus wasting energy on probes that may fail. PISTONv2 performs particularly well when the speed changes are gradual. However, if a node is static and then starts to move, then the computed IPT is large for slow speeds. Thus, similar to eDiscovery, PISTONv2 may miss some of the contacts at the beginning of motion.

Relative to eDiscovery, PISTONv2 adds speed detection complexity to each probe. While the speed detection may cause an additional energy drain, we expect that the approximate values of speed will suffice. Thus, we can infer the speed from low-power sensors such as accelerometers, gyroscopes or other coarse location sensors. Even fluctuations in radio signals can be used to detect mobility, as in the case of DWARF [74].

6.2.3. *STAR*

Based on the number of encounters experienced by the node over the past 24 hours (e.g., hourly), STAR [70] calculates the IPT for the subsequent time slot. Since time-of-day information is used, this approach is referred to as STAR-TOD. Authors have observed that the pattern may have different biases on different days of the week, subsequently using the arrival rate in the previous time slot (PTS) to calculate the arrival rate of the current slot. STAR-PTS calculates the probability of missing encounters using the IPT value from the previous slot and the number of encounters for the previous slot. STAR estimates the number of contacts (λ) for next slot, using details from the last slot that had contacts. Eq. 6.2 is used to compute IPT, given τ (minimum target IPT), the Pareto distribution constant k (0.85 in [70]), the cost factor c and the estimated number of contacts λ.

$$IPT_{PTS} = \tau\left(1 + \frac{c(1-k)}{\lambda\ k\ \tau}\right)^{1/(1-k)} \quad . . . . \text{Eq. 6.2}$$

In their original work, the authors set the value of STAR_{PTS} using the STAR_{TOD} value in the morning, to avoid the warm-up time typically associated with the absence of contacts during the night. To compare our work with STAR, we use a weighted value of λ, as given in Eq. 6.3. λ_{TOD} is the estimated arrival rate based on the number of contacts encountered 24 hours earlier, w is a weight factor between 0 and 1, and λ_{PTS} is the estimated arrival rate based on the number of contacts in previous time slots.

$$\lambda = w * \lambda_{PTS} + (1-w) * \lambda_{TOD} \quad . . . . \text{Eq. 6.3}$$

STAR is slightly more complex than eDiscovery, as it needs data regarding historical contacts, including data from prior days. This adds an order of a few bytes per node regarding storage complexity. The computational complexity is negligible, as the IPT calculations are done only once per slot.

6.2.4. *Motivation for STEEP*

Although STAR uses the previous slot or TOD to optimize the IPT values and demonstrates better discovery rates, simulation has shown that the number of probes is significantly higher if the node densities increase. PISTONv2 achieves lower probe rates as it computes very long IPT values for static nodes. Since higher-speed mobile nodes have a shorter IPT, they drive most of the discovery.

Figure 6.5 on page 117, shows the duration spread of contacts and time interval between speed change to the discovery of contact for continuous probing (every 100 milliseconds) across three simulation scenarios. After studying the performances of eDiscovery, STAR, and PISTONv2, we explored multiple approaches to improve the trade-off between discovery and energy. Based on the observation that missing short-term contacts (below 1-2 seconds in Figure 6.5(a)) do not have a significant impact on DTN routing, we performed multiple experiments to improve the discovery rates. We found that when a neighbor node sends a probe to discovers a static node, the static node likely has several more discoverable nodes in its vicinity. For such scenarios, discovery can improve, if the static nodes reduce IPT by reacting to connection-up events triggered by the neighbor nodes. Figure 6.5(b) shows that more than 60% of contacts are discovered within the first minute after movement starts. Even SFT with cars driving for long duration discovered 57% percent of contacts within one minute. In all cases, less than 2% of contacts were discovered 60 minutes after the start of the movement. This analysis shows the importance of tracking the node speed changes, which made us include a parameter for speed check in STEEP. To conserve energy beyond the probes, we also explored turning off the receiver between the probes, as discussed in Section 4.

6.3. STEEP – Speed and Time-based Energy Efficient Probing

STEEP uses an approach similar to that of PISTONv2 with two extensions. The first extension increases the IPT in the absence of new contacts and in the absence of an increase in node speed. The second extension periodically checks for a speed increase, at which point it resets T_{lse} and re-computes IPT, as captured in Algorithm 3.

STEEP computes the IPT according to Eq. 6.4. R is the radio range, and B is the current node speed. T_{lse} is a variable that captures the time since the last significant event (LSE). LSE is either a connection-up event (captured using Algo 6.4) or a significant increase in the node speed. In addition to $T_{lse,}$ the two constants $LSE_{penalty}$, and AF_{disc} are included.

$LSE_{penalty}$ is the time penalty that determines the weight given for LSE. AF_{disc} is the aggression factor for discovery.

$$IPT = (R + T_{lse} * LSE_{penalty})/(AF_{disc} * B) \quad \text{Eq. 6.4}$$

Speed (B) and T_{lse} are the only two variables needed to compute the IPT. R is the radio range, which depends on the node's capability. $LSE_{penalty}$ and AF_{disc} are algorithm constants determined based on the deployment goals. The equation shows that higher $LSE_{penalty}$ values help reduce the amount of energy spent on probes by increasing the time between probes. A higher AF_{disc} value reduces the inter-probe time and thus expends more energy, but it also reduces the chance of missing contacts.

Eq. 6.4 provides a satisfactory trade-off between the density of nodes and the number of probes. When the number of nodes is small, the gap between new contacts is large. Thus, a speed increase is the primary contributor for LSE. In such cases, $T_{lse} * LSE_{penalty}$ computes low IPT values predominantly after a speed change. As the node count increases, nodes encounter more frequent contacts; thus, connection-up events dominate LSE, and T_{lse} typically stays low. For low T_{lse} values, the radio range R drives the gap between probes, thereby ensuring that the IPT is never below $R/(AF_{disc}*B)$. Both R and AF_{disc} are constants, and the node speed B is not affected by the density. Thus, this approach scales well relative to STAR and eDiscovery. Section 4.5 provides a detailed mathematical analysis of PISTONv2 and STEEP.

6.3.1. *Probe and IPT computation*

Algorithm 6.2 STEEP – SendProbeAndComputeNextIPT

```
if current_time > (lastProbeTime + computedIPT) then
    Probe for Neighbors;
    B = getCurrentNodeSpeed();
    t_since_lse = CurrentTime - time_lse;
    computedIPT = (R + t_since_lse * LSE_penalty) / (B * AF_disc);
    Ensure computedIPT is within valid range ;
    lastSpeedChecked = lastProbeTime = CurrentTime;
end if
```

STEEP computes inter-probe time, after sending the probe, as detailed in Algo 6.2. To avoid dividing by zero, the speed is set to 0.01 m/s for static nodes.

6.3.2. *Periodic speed check*

After a probe has been sent, the nodes periodically check (Speed_ChkIntrvl – default = 1 minute) if the speed has increased significantly. If the speed increases by more than the Speed_Factor (default is 10), the IPT value is divided by the SpeedFactor. This algorithm is only utilized if the ComputedIPT is larger than Speed_ChkIntrvl and affects IPT only if the speed increases significantly.

Algorithm 6.3 STEEP – PeriodicSpeedCheck

```
if current_time > (lastSpeedChecked +Speed_ChkIntrvl) then
lastSpeedChecked = current_time;
newB = getCurrentNodeSpeed()
if newB > Speed_Factor * B then;
        B = newB;
        computedIPT = computedIPT / Speed_Factor;
        time_lse = current_time;
end if
end if
```

6.3.3. *Tracking “connection up” events*

Whenever a node detects a “connection up” event, it adjusts the “time_lse”. Note that for Node A, the “connection up” event can be triggered after a new node responds to a probe sent by Node A or if Node A receives the probe from another node. It just sets the time_lse to current_time.

6.3.4. *Algorithmic complexity of STEEP*

It is assumed that nodes are capable of measuring their speed with the help of an accelerometer, a GPS or similar approaches. “SendProbeAndComputeNextIPT” (probe and IPT computation) does not experience any significant change compared to PISTONv2 for either computation time or storage space.

“PeriodicSpeedCheck” does add to computation complexity when compared to the other algorithms, but it is applicable only when IPT values are large (about a minute or more). It is expected that mobile nodes will implement hibernate logic such that they are active for a brief period every few minutes to implement these checks even when there is no node in their vicinity.

“OnConnectionUp” does not really add to the complexity, as a node is supposed to react to these events and process information for the contact. Only additional processing for STEEP is to set the value of time_lse to the present time of the node.

6.4. Energy consumption in wireless protocols

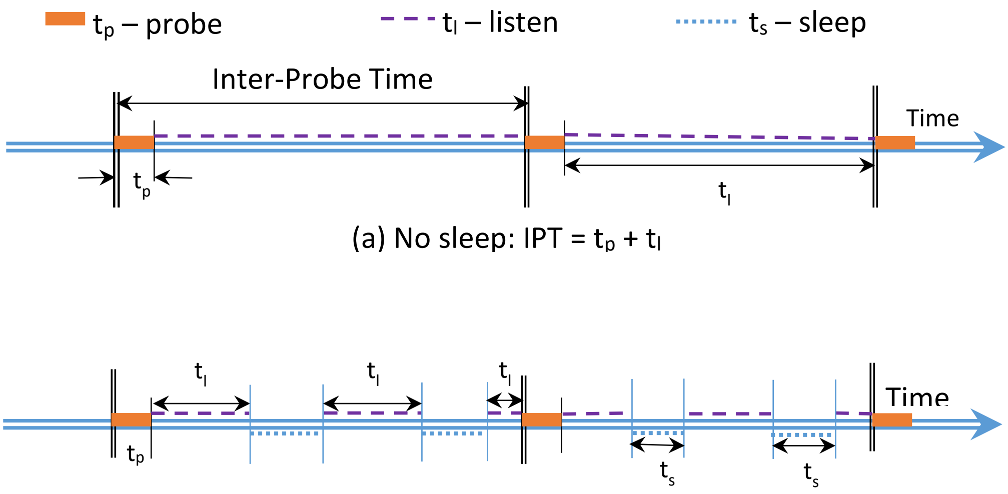
As described in [75], a typical mobile node undergoes four stages of operation: transmit, receive, idle and sleep. In the transmit mode, nodes send messages to neighbor nodes that are within range (e.g., each node sends a probe or beacon for discovery, sends DTN routing information or forwards application bundles). In receive mode, the nodes activate their receive circuitry to obtain messages or probe responses from neighbors. In idle mode, nodes listen to the channel primarily to sense carrier occupancy and then determine a suitable channel for message transfer. The decode and process circuitry of a receiver is not completely engaged in idle mode, but it can receive and process probe requests. In sleep mode, the nodes enter a low-power mode, thus they are not involved in data transfer or discovery activities. Depending on the type of device, the sleep mode may completely turn off the receiver, or the receiver may wake up for very short durations.

Of the four operating modes, nodes may stay in idle mode most of the time during the peer discovery process [75]. On a cumulative basis, this mode consumes more power than the other modes. Another work by Feeney et al. [76] presents the power consumption of 802.11 devices. Their work shows that the power consumption of sleep mode is ten times lower than that of idle mode and other modes. While past devices may not have had a provision for the sleep state, current devices are increasingly deploying advanced power management strategies such as hibernate and deep sleep to conserve power. Zhang et al. [77] proposed the cognitive sensing of channels to optimize the power consumption in small cell networks at physical and link layers. This optimization can provide additional power savings in addition to the savings achieved by adapting the IPT or using the sleep state. To ensure that our framework can accommodate heterogeneous network interfaces, we do not assume any such link layer optimization. The energy savings from these link-layer optimizations are complementary to the savings from discovery algorithms.

eDiscovery, PISTONv2 and STAR assume that remote nodes do not enter sleep mode but rather listen to neighbors to receive probe requests. Previous observations [75] have revealed that in sparse networks, the idle mode contributes to 95% of the total energy consumption. The objective of the neighbor discovery algorithm is to reduce the number of probes, and the energy spent in listen mode. Here, we evaluate the performances of different discovery algorithms when the nodes intermittently operate in sleep mode. In this section, we present an overview of the power-saving mechanism and its expected impact on discovery. Subsequently, we present an overview of the energy spent on Bluetooth and Wi-Fi (802.11) protocols and study the energy consumption in various modes. The next subsection discusses the impact of the four algorithms on the discovery concerning the discovery failures if other nodes are in the sleep state. Finally, we provide a mathematical analysis to compare the performance of PISTONv2 and STEEP.

6.4.1. *Power conservation between probes*

In a conventional operation, during the inter-probe interval, a node must be in listen mode to receive and respond to a neighbor probe request. As discussed in the introduction, on a cumulative basis, listening mode consumes significantly more power than the other modes. Jun and others [78] discuss algorithms for sleep scheduling in the listening window between probes. Their work shows that if nodes do not have any prior knowledge of the sleep state of other nodes, then they tend to use maximum energy to send probes. We attempt to evaluate the performances of various discovery algorithms under this scenario.



(a) No sleep: IPT = $t_p + t_l$

(b) Periodic sleep between probes: IPT = $t_p + \sum t_l + \sum t_s$

Figure 6.1 Inter-probe time, probe time, sleep and listen modes

Figure 6.1 provides an example for nodes operating in various modes such as probing, listening, sleep and IPT duration. In Figure 6.1(a), the node may either send a probe (t_p) to other nodes or listen for probes (t_l) from other nodes during the IPT. Figure 6.1(b) adds periodic sleep (t_s) intervals during the listening time.

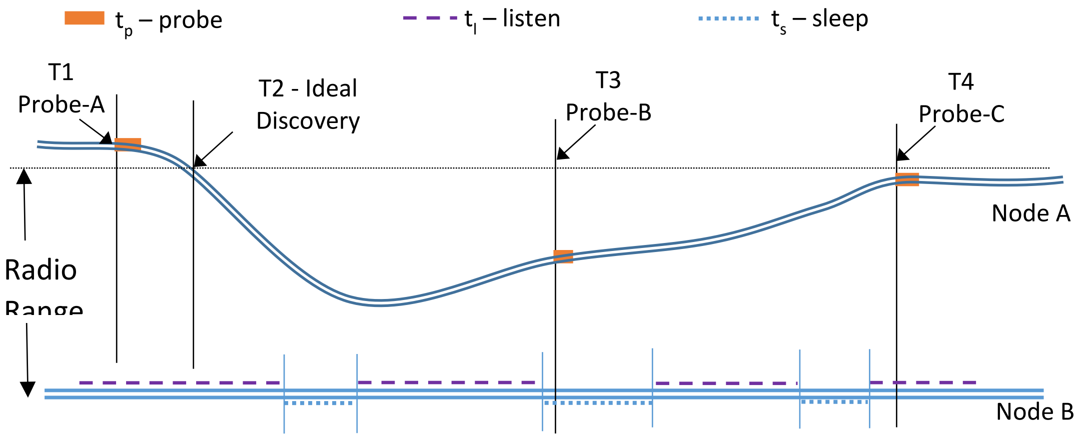


Figure 6.2 Efficiency of probe with distance and neighbor sleep pattern

Figure 6.2 captures the distance between two nodes and the sleep pattern for node B. For simplicity; we assume that node B has a relatively large IPT value; thus, this node sends no probes. Only the relative distance between node A and node B with respect to the Y-axis is presented in the figure. During probe-A at T1, node B is outside the radio range of node A, does not receive the probe and cannot be discovered. If continuous probing is performed (i.e., the ideal case), discovery is expected to occur at T2. During probe-B at

T3, node B is sleeping and does not respond to the probe. Subsequently, the discovery of node B by node A via probe-C occurs at T4 (i.e., when node B is within the radio range and is in listen mode).

Events such as missed discovery at T3 cause contact durations to decrease or some of the short contacts to be missed. For example, if the radio range were slightly smaller, no contacts would be detected in Figure 6.2. Probe-A and probe-C would fail since the distance is too large, and probe-B would be missed because node B is in sleep mode.

Note that even if the other nodes are not sleeping, because we do not perform continuous probing, the duration of contact decreases, and some of the short contacts may be missed. We explore this topic more thoroughly when we present the analysis of the results in Section 6.

6.4.2. *Sleep scheduling mechanism in simulation*

We simulated sleep by turning off the receiver radio during the inter-probe time. A node will undergo periodic sleep and active cycles for the computed time duration. The sleeping duty cycle can be varied based on the required network performance and energy conservation goals. An alternate approach to cyclic sleep between probes is to perform continuous sleep in one stretch to achieve the required power reduction. We chose to implement sleep cycles similar to Figure 6.1(b), without any synchronization of the sleep durations between nodes.

As expected, simulations done using different sleep targets demonstrated that the battery could last almost twice as long when the node sleeps 60% of the time. In section 6, we cover the impact on discovery of sleep-based energy conservation targets.

6.4.3. *Power consumption*

Wireless chipsets such as CC2541 [79] and CC3200 [80] support sleep mode or low power mode, in which the receivers can be made inactive. The TI chipset CC2541 was developed for BLE (Bluetooth low energy) applications, and TI CC3200 supports 802.11

b/g/n protocols. These chipsets are designed with inbuilt power management features for energy harvesting applications to enhance the battery lifetime.

In active mode, the CC2541/CC3200 chipset undergoes protocol-specific events (advertise, probe, page, connect, etc.) and application-specific events (Tx/Rx). In power mode 1, the CC2541 chip turns off the processing core and RF interfaces, and in power mode 2, the device is turned off entirely. Similarly, for CC3200, the micro-controller and RF core are turned off in LPDS (low-power deep sleep) mode, and the entire chip is turned off in hibernate mode.

[79] and [80] discuss the experimental setup and power measurement procedures for these chips for various power profiles. Table 6.1 provides the current and power consumption for CC2541 (3 V) and CC3200 (3.6 V), which were obtained from the TI datasheets. These values are used in Eq. 6.5 to determine the inputs for the simulations.

The power consumed in low-power deep sleep mode is ten to one hundred times lower than that in listening mode. Hence, if the nodes decide to conserve energy by entering LPDS on a periodic basis, they can significantly extend their battery life.

Table 6.1 Power consumption for discovery

Mode	CC2541 BLE 4.0		CC3200 Wi-Fi	
	Current [mA]	Power [mW]	Current [mA]	Power [mW]
Transmit	18.2	55.8	229	824.4
Receive / Listen	17.9	62.4	59	212.4
LPDS	0.27	0.81	0.25	0.9
Sleep	0.001	0.003	0.004	0.0144

To compute energy spent on discovery, formulae from [81] is simplified for our work, as shown in Eq. 6.5(a) to (f). The key notations used in the above equations are summarized in Table 6.2. Since T_{Probe} and $T\_{ProbeResp}$ are very small, they are considered insignificant concerning the IPT.

$$\left.\begin{aligned}
&E_{\_Total} = E_{\_Probe} + E_{\_Listen} + E_{\_Sleep} \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; && \text{(a)}\\
&E_{\_Probe} = (P_{\_Tx} * T_{\_Probe}) + M*(P_{\_Tx} * T_{\_ProbeResp}) \;.\;.\;.\; && \text{(b)}\\
&E_{\_Listen} = (P_{\_Listen} * T_{\_\ Listen}) \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; && \text{(c)}\\
&T_{\_\ Listen} = (1\text{-}SP) * IPT \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; && \text{(d)}\\
&E_{\_Sleep} = (P_{\_Sleep} * T_{\_Sleep}) \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; && \text{(e)}\\
&T_{\_Sleep} = (SP) * IPT. \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; && \text{(f)}
\end{aligned}\right\} \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; \text{Eq. 6.5}$$

Table 6.2 Notations used in the energy model

Notation	Description
$E_{\_Total}$	Total energy consumed by a node for discovery
$E_{\_Probe}$	Energy consumed to send and respond to probe requests
$E_{\_Listen}$	Energy consumed in idle listening mode (Receive / Listen in Table 1)
$E_{\_Sleep}$	Energy consumed in sleep mode
$T_{\_Probe}$	Time duration of transmitting a probe
$T_{\_ProbeResp}$	Time duration of transmitting a probe response upon receipt of probes by others
$T_{\_Listen}$	Total time spent in idle / listen mode
$T_{\_Sleep}$	Total time spent in sleep mode
M	Number of probe responses sent during the IPT
SP	Sleep percentage – % of time spent in sleep mode

6.4.4. *Impact of sleep on various discovery mechanisms*

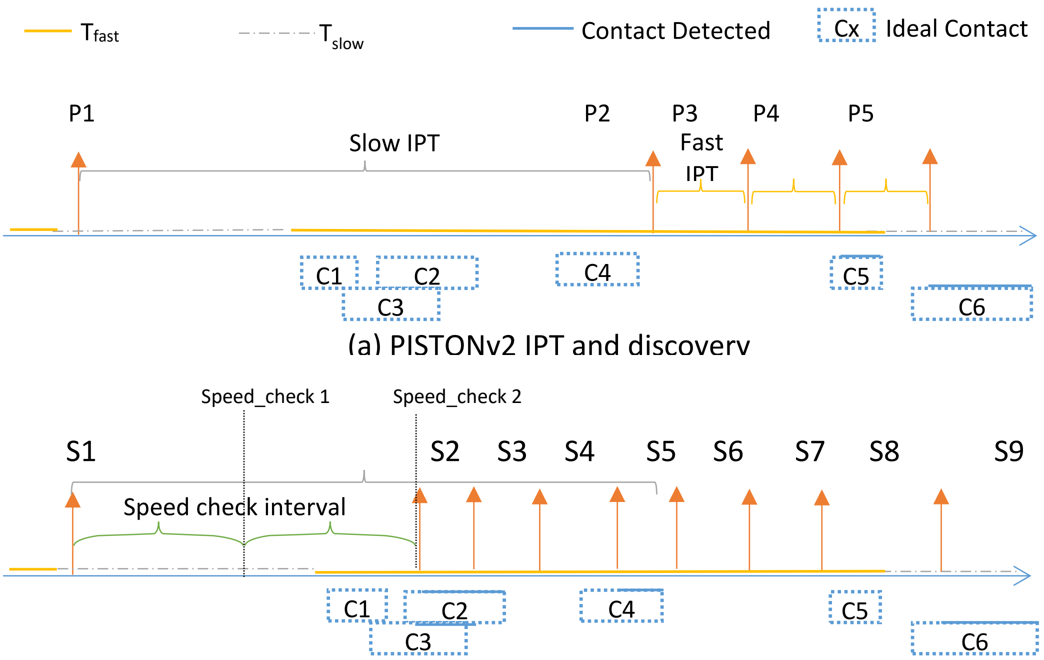
Table 6.3 summarizes the effects of sleep IPT and its subsequent impact on discovery in the four discovery algorithms. Note that as nodes sleep longer with larger values of sleep percentage (SP in Eq. 6.5), they do not respond to probes from other nodes, thus reducing the discovery. Ideally, the reduction is linear to the SP.

Table 6.3 Impact of sleep on discovery algorithms

Discovery Algorithm	Impact to the IPT on missed discovery because the other node is in the sleep state	Impact on discovery effectiveness for larger SP
eDiscovery	Increases the IPT because of logic in line 4 / line 10 of Algo 6.1. Resets the IPT to a small value as soon as any discovery succeeds (line 6 of Algo 6.1).	A larger IPT causes additional contacts to be missed. This condition is less favorable than linear degradation as nodes sleep for longer times.
PISTONv2	No impact on IPT since Eq. 6.1 is independent of contact success.	The ratio of missed contacts is similar to the ratio of sleep time; thus, linear degradation is expected as nodes sleep for a longer interval.
STAR	Causes lower (λ) estimates for subsequent probes. This increases the IPT (Eq. 6.2).	Since subsequent contacts do not reset (λ), this condition is less favorable than linear degradation as nodes sleep for longer targets.
STEEP	Missed contacts affect the T_{lse}, making it larger and thus slightly increasing the IPT. The impact is reset on the next discovery or next speed change by the node.	Linear degradation occurs as long as T_{lse} * $LSE_{penalty}$ is smaller than radio range R in Eq. 6.4. A condition less favorable than linear degradation occurs in the absence of a speed change coupled with low node density.

6.4.5. *Probe energy and discovery tradeoff*

In this sub-section, we analyze the STEEP algorithm and its energy conservation mechanism. For discovery, energy consumption is directly proportional to the probe counts; thus, we focus on the probe counts and discovery in the trade-off analysis. As captured in Eq. 6.1 and Eq. 6.4, in the absence of significant events, our approach is similar to that of PISTONv2. Therefore, we explore the energy trade-off concerning PISTONv2. For ease of discussion, let us assume that nodes do not sleep and that they move at only two speeds (B_{fast} and B_{slow}). Let M denote the ratio of B_{fast} (the fast speed, corresponding to car or bus travel) to B_{slow} (the slow speed, corresponding to a static node or pedestrian mobility).



(a) PISTONv2 IPT and discovery

(b) STEEP IPT and discovery

Figure 6.3 Example of probes and discovery for PISTONv2 vs. STEEP

Figure 6.3 illustrates the probes and discovery for PISTONv2 and STEEP. PISTONv2 uses fewer probes (P1-P5) but also misses many of the contacts. When the probe is sent at P1 with speed $B_{slow,}$ PISTONv2 computes a large value of IPT, thus resulting in the probe being only at P2. Thus, this probe misses the intermediate contacts C1-C4. In contrast, STEEP performs periodic speed checks, as shown in Figure 6.3(b). The first speed check (designated speed check 1) does not cause a change because the node is still in slow mobility mode. If M > (Slow_IPT) / (Speed_check2 – S1), then the second speed check triggers an immediate probe at S2. If M is smaller, the next probe may occur following a delay, but the gap of P1-P2 will be much larger than the gap of S1-S2 (since M > 1).

In this example, the detection of the change in speed and the IPT update by STEEP (Algo 6.3) increases the discovery rate. However, the detection also causes STEEP to send nine probes as opposed to the five probes in PISTONv2. The benefits of additional discovery in STEEP depend on the probability of the encounter occurring after the increase in node speed. Similarly, the cost of probes is high when T_{lse} is small. T_{lse} is set to zero under two conditions: after the speed-change detection and after the connection-up events. We now analyze the criteria under which the cost of discovery of STEEP is better or worse than that of PISTONv2. Figure 6.5(b) captures the probability of discovery from the time of the detection of the last speed change for all three mobility models using continuous probing. The plots follow a power law distribution with its characteristic long-tail pattern. The majority of the contacts occur within a minute of the speed change. For the two-speed scenario discussed above, Eq. 6.6 models the number of contacts missed within a minute of the increase in speed by PISTONv2. Let NumSC denote the number of times the speed changes from F_{slow} to F_{fast}. P_{sc} is the probability of a contact occurring within a minute of a speed change. K is a constant representing the ratio of the contacts that are likely to be discovered by probes sent with a gap of IPT_{fast} to the contacts discovered under ideal conditions. For example, if all contacts are of 1 second, when IPT_{fast} is 10 seconds, K is set to 0.1. K is a constant that depends on the scenario and has an upper bound of 1.

$$\text{MissedByPistonV2} = NumSC * P_{sc} * totalContacts * K \text{ . . . . Eq. 6.6}$$

Since STEEP detects the speed change only at the next speed check interval, this approach misses some of the contacts in the early part of the movement, such as C1 in Figure 6.3(b). On average, using a speed check interval of one minute, STEEP typically captures only half the contacts. Eq. 6.7 shows the improvement in discovery for STEEP immediately after movement starts. In the absence of significant events (such as speed changes or new connections), STEEP increases the IPT because of T_{lse} in Eq. 6.4. Since $LSE_{penalty}$ is small, this condition is relevant only after a few minutes. Given that contacts follow the long-tail pattern, one can ignore the contacts missed because of the increase in IPT after several minutes.

Disc. improvement by STEEP $= NumSC * P_{sc} * totalContacts * \frac{K}{2}$ . . . . Eq. 6.7

Note that we have used one minute as the interval after the speed change for simplicity of discussion; typical values may be larger. For intervals larger than a minute, the P_{sc} value is higher. Eq. 6.7 provides a lower bound for the improvement.

When comparing the frequencies of probes by PISTONv2 and STEEP, we observed that the speed-change detection sends additional probes, while T_{lse} * $LSE_{penalty}$ saves probes by increasing the IPT value. We first identify the additional probes that result from speed-change detection by ignoring the impact of the last significant event. Eq. 6.8 captures the number of additional probes because of the speed change in STEEP (Algo 6.3). Since the periodic speed check occurs at a discrete interval and a speed change could have occurred at any time prior, Eq. 6.8 gives the upper bound of the number of probes.

Max. additional probes for speed change in STEEP $= NumSC * M$ . . . . Eq. 6.8

PISTONv2 probes $= \frac{Tslow}{IPT_{slow}} + \frac{Tfast}{IPT_{fast}} = \frac{T_{slow}*2*\alpha*B_{slow}}{R} + \frac{T_{fast}*2*\alpha*B_{fast}}{R}$ . . . . . . Eq. 6.9

By substituting B_{fast} = M * B_{slow}, we obtain

PISTONv2 probes $= \frac{2*\alpha*B_{slow}}{R}(T_{slow} + M * T_{fast})$ . . . . Eq. 6.10

A similar approach for STEEP (with AF_{disc} set to 2α) gives the number of probes (not counting the impact of the change in speed) as

STEEP probes without speed change $= \frac{2*\alpha*B_{slow}}{R+J}(T_{slow} + M * T_{fast})$ . . . . Eq. 6.11

where J denotes the average value of T_{lse} * $LSE_{penalty}$.

Savings on LSE $= \frac{2*\alpha*B_{slow}}{R}(T_{slow} + M * T_{fast})$ - $\frac{2*\alpha*B_{slow}}{R+J}(T_{slow} + M * T_{fast})$ . . Eq. 6.12

Savings on LSE $= \frac{2*\alpha* B_{slow}*J}{R*(R+J)}(T_{slow} + M * T_{fast})$ . . . . Eq. 6.13

For STEEP to be more energy efficient than PISTONv2, the number of probes should be lower. Thus, by combining Eq. 6.8 and Eq. 6.13, the criteria for STEEP to be better than PISTONv2 on energy savings is

$2 * \alpha * B_{slow} * J(T_{slow} + M * T_{fast}) > NumSC * R * (R + J) * M$ . . . . Eq. 6.14

By removing the constant 2*α, the benefits can be captured as an improvement in the ratio given by Eq. 6.15.

$STEEP\ Probe\ Improvement\ Ratio = \frac{B_{slow}*J(T_{slow}+ M*T_{fast})}{NumSC*R*(R+J)*M}$ . . . . Eq. 6.15

Eq. 6.7 and Eq. 6.15 give the criteria for which STEEP outperforms PISTONv2 both in discovery and energy conservation. The discovery improvements are proportional to NumSC * P_{sc}. Eq. 6.15 shows that high values for NumSC triggers additional probes, thereby leading to additional energy consumption. Thus, for STEEP to be effective on both discovery and energy, P_{sc} should be high. Since most of the mobility scenarios follow the long-tail pattern, P_{sc} is high in such cases. Figure 6.5(b) also demonstrates that 60-80% of contacts occur within the first minute of the speed increase.

If we target only energy efficiency, STEEP is impacted by 1) the radio range R, 2) NumSC, 3) B_{slow}, 4) the LSE component J, 5) the speed multiple M, and 6) the ratio of T_{fast} to T_{slow}. For devices with small radio ranges, STEEP provides significant benefits since the factor R*(R+J) is small. For scenarios featuring frequent speed changes (higher NumSCs), STEEP performs more poorly than PISTONv2 in terms of energy. Note that a low value of B_{slow} reduces the savings for STEEP. Since there is a fixed upper bound on the computed IPT (MaxIPT), the impact of B_{slow} is mitigated to some extent when nodes stay static.

If J (i.e., the LSE component in both the numerator and denominator of Eq. 6.15) is too small, STEEP does not save energy. However, if J increases to a value much higher than R, then the drawback is diminished since the term of (R+J) in the denominator of Eq. 6.15

tends to be J (if J >> R), thus effectively negating the impact of J in the numerator. However, a high value for J reduces the discovery by increasing IPT. Thus, J (the average value of Tlse * $LSE_{penalty}$) should be controlled to have a similar value as R.

The speed multiple M and the ratio of T_{fast} to T_{slow} exhibit a relatively complex relationship. A high value of M affects both the numerator and denominator. Thus, the influence of M depends on the ratio of T_{fast} to T_{slow}. When T_{fast} is insignificant relative to T_{slow}, for large values of M, STEEP is not efficient in saving energy. This result is particularly prominent the WDM-fast scenario, in which nodes move at high speeds for short durations. In contrast, if nodes are mobile most of the time (e.g., real-world taxi traces), the influence of M balances out in both the numerator and denominator of Eq. 6.15.

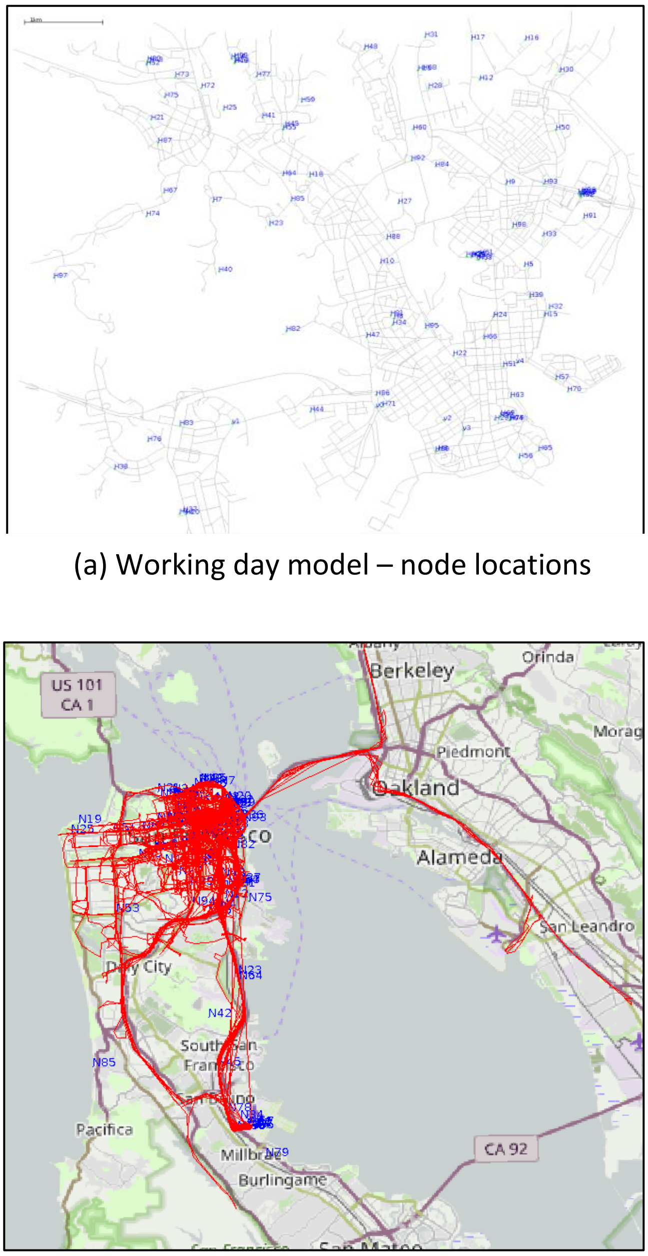
6.5. *Experimental Setup*

The opportunistic network environment (ONE) simulator [41] was used with the synthetic mobility working day model (WDM), as described in [42], and real-world taxi traces from SFT datasets [43]. Other than eDiscovery, PISTONv2, STAR and STEEP, the discovery was also performed via ideal and constant interval probing. *Ideal* discovery sends continuous probes to ensure that no contacts are missed during simulation. Constant interval probing (referred to as *Const* in the rest of the chapter) sends probes at fixed intervals.

Table 6.4 Simulation settings

Variable	Slow / Fast	SFT
Path Layout / Map	Helsinki City Map	San Francisco Taxi Traces
Movement Model	Synthetic - WDM	Trace-Based
Map Size	10 km x 8 km	52 km x 100 km (approx.)
Number of Nodes	100 (default)	
Wireless Range	10 m	
Wireless Speed	48 mbps	
Simulation Duration	144 hours (24-hour warm-up)	

Table 6.4 captures the simulation scenario settings. For WDM-slow, nodes moving at pedestrian speeds (0.8-1.4 meters/second) or by bus (7-10 meters / second). For WDM-fast nodes use car (7-30 meters/second) or bus. By default, out of 100 nodes, five nodes represented buses, and the remaining 95 nodes followed either the fast or slow working day mobility.



(a) Working day model – node locations

(b) San Francisco taxi trace – node path

Figure 6.4 Simulation – graphical user interface screenshots

The SFT[43] dataset was cleaned up using steps identified in section 3.3.3. For multiple runs, we selected traces corresponding to 4 different weeks and used data from 100 most active taxis for that week.

Figure 6.4(a) presents the screenshots of the WDM and the node distribution in the network. Figure 6.4(b) shows the SFT map and the mobility traces of an individual cab for an entire week.

6.5.1. *Energy model*

ONE code is extended to support sleep mode on the network interfaces without triggering a link-down event. During sleep mode, no new connections can be created even if other (non-sleeping) nodes send discovery probes. Interfaces sleep in bursts of 500 milliseconds. They exit sleep mode before the next probe. During experiments, the target sleep was varied from 0% to 60% in steps of 10%.

Table 6.5 Energy model

Discovery Mode	Wi-Fi Interface
Initial Energy [mA-s]	30,000,000
Probe Energy [mA-s]	76.0
Probe Response Energy [mA-s]	63.25
Listen Energy [mA-s]	59.0
Sleep Energy [mA-s]	0.25

Table 6.5 presents the energy model used in the simulation. The initial energy levels ensure that the battery does not drain within five days. Other values are computed using Eq. 6.5 and the values from Table 6.1.

6.5.2. *Variants for simulation execution*

Each scenario was executed with one of four different randomization seeds to remove bias. Similarly, for the SFT scenario, the location traces corresponding to four different weeks were used to reduce bias. The results presented in the next section are the averages for the four runs. Most of the results are within 2-10% of the average value for the 95% confidence interval. STAR with its time-of-day needs some interval for learning. Hence, the first-day simulation data was discarded for all runs to skip the learning period.

Table 6.6 Static and mobility distribution for a typical node across scenarios

	Min	Max	Avg
Slow	63.02%	95.80%	83.62%
Fast	81.46%	99.54%	94.76%
SFT	2.22%	83.25%	16.49%

Table 6.6 shows the percentage of samples when the node is static (movement of less than 0.1 meters in 10 seconds). In the fast scenario, the nodes completed their travel quickly, while they took longer in the slow simulation. This caused the WDM-fast nodes to be stationary for longer durations; hence, STEEP and PISTONv2 are expected to use fewer probes. For the SFT scenario, the average number of samples that were static dropped significantly, as the taxis were mobile most of the time while serving their customers.

Table 6.7 Values for Const, eDiscovery, PISTONv2, and STAR

Algorithm	**Parameter**	**WDM**	**SFT**
Const	IPT	30	1.0
eDiscovery	inc_NP	1	0.01
	I	0.1	0.001
	base_I	3.7	0.37
STAR	τ	3.7	0.37
	k	0.85	0.85
PISTONv2	α	1	1

More than half (60%) of the contacts were separated by 1024 seconds. Hence, the value of MaxIPT was set to 1024 seconds for all algorithms. Details regarding the simulation constants are presented in Table 6.7 for Const, eDiscovery, PISTONv2, and STAR. Since [72] used τ = 3.7 for STAR, we also used this value for the slow and fast simulations. The remaining values were identified to ensure that the unique discovery ratios on a daily basis were above 50% of the ideal value.

6.5.3. *Scalability characterization*

Historically, opportunistic networks have been associated with low-density deployments. In [82], the authors observe that these networks may make sense even for higher-density deployment. Density-Based performance evaluation has been attempted for DTN routing, e.g., [83, 84].

To understand how the algorithms behave under different node densities, we increased the number of nodes from 25 to 1600 for the same map area in WDM scenarios. The number of buses was varied non-linearly from 2 to 25 nodes. The SFT traces were not used for these simulations because as we increased the number of taxis beyond 100, many of the nodes remain parked in the taxi stand for long intervals, thus skewing the results.

6.5.4. *STEEP variation analysis*

Table 6.8 STEEP algorithm – Simulation values

	Default / Base runs	Isolated runs	Variations
AF_{DISC}	1	2.0	0.5 – 2.0
SPEED_CHKINTRVL	60	60	1 – 300
SPEED_FACTOR	10	10/90,000	1.25 – 1280
$LSE_{PENALTY}$	0.001	0	0.01 – 0.05

Table 6.8 presents the default values used in the STEEP simulation runs. Also, to evaluate the contribution of each algorithmic component, we executed simulations of STEEP with variations, as shown below. For base runs, only one value was varied at a time, with the rest retaining their default base values. For isolated runs, the intent was to

capture the individual contribution when other algorithmic components were not contributing to the algorithm. For example, when Speed_Factor is set to 90,000, a significant event will never occur because of the change in speed.

6.5.5. *Impact on routing*

Binary spray-and-wait (SNW) [14] DTN routing is used for these experiments. As we varied the SNW replica count from 1 to 128 in power of 2, we were able to better observe the network tradeoffs. With the replica count set to 1, SNW becomes more like direct delivery [85]. Similarly, for a count of 128, SNW degenerates to epidemic routing [86] with larger overheads. On an average, messages of 1 MB are generated every 60 seconds with time-to-live (TTL) of 24 hours. The nodes had 1 GB buffer space.

6.6. Results and performance evaluation

From the simulation results, the following metrics are used for the performance evaluation:

- *Number of unique contacts or discoveries* – Suvadip and others [87] stated that unique contacts are dominant in improving the performances of DTN routing protocols. Hence, instead of the total number of contacts, we used the number of unique contacts as the metric for evaluating performance. The plots of the unique contacts in Figure 6.6(a, b, c), Figure 6.9(a, b), Figure 6.11(a, c, e, g) and Figure 6.12 (a, c, e, g), were normalized as a percentage of the unique contacts relative to the ideal discovery (i.e., an ideal run would achieve a score of 100%).
- *Energy spent to send probe requests* – The number of probes sent was ten to fifty times higher than the number of probe responses. For simplicity, we used the number of probes sent as the metric for energy spent in discovery. The values in Figure 6.6(d, e, f), Figure 6.9(c, d), Figure 6.10(c, d), Figure 6.11(b, d, f, h) and Figure 6.12(b, d, f, h) were normalized using the number of probes that would be sent if constant interval probing was used. Note that we cannot use normalization against the ideal discovery for this metric because the ideal discovery approach conducts continuous probing.

- In addition to the above metrics, we capture the *contact duration spread*, in Figure 6.7 for different discovery algorithms and analyze the impact on the *delivery ratio* for spray-and-wait routing in Figure 6.8.
- For sleep-based simulations, we plot the *number of unique contacts* and *normalized probes* for various discovery algorithms for different target sleep percentages in Figure 6.10.

As discussed before for unique contacts and the number of probes, the average values from four runs were used. Moreover, to avoid the skew from the initial adaptation, Day-0 (i.e., the first 24 hours) data were excluded from the results.

6.6.1. *Baseline behavior of contacts*

Figure 6.5 captures the contact duration and time between last speed change to start of contact under ideal conditions (continuous polling). In essence, Figure 6.5(a) represents the best-case scenario for contact duration spread. Figure 6.5(b) shows that majority of the contacts happen within a minute of speed change (as discussed concerning analysis in sub-section 6.4.5).

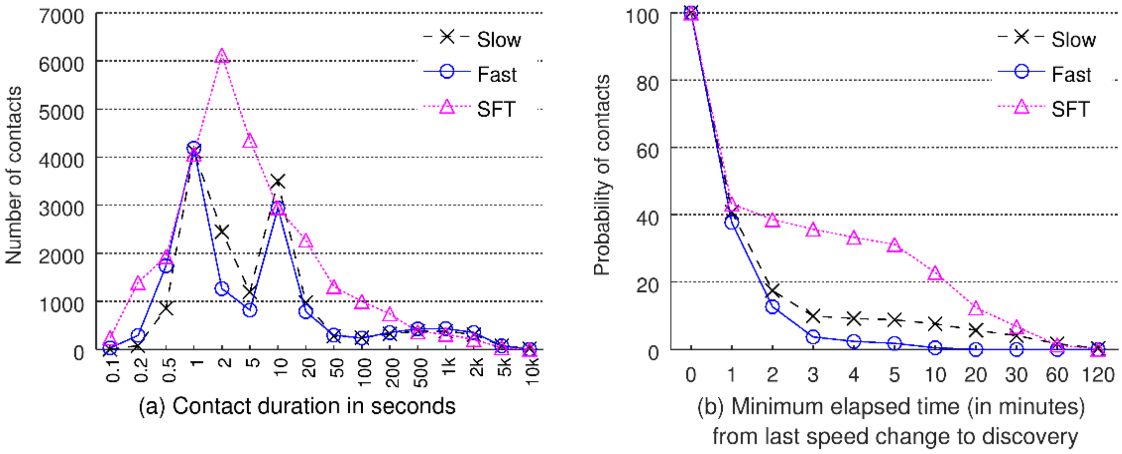


Figure 6.5 Contact duration and discovery times across scenarios

6.6.2. *Performances for unique discovery and number of probes*

Figure 6.6 captures the percentage of unique contacts for Const, eDiscovery, PISTONv2, STAR, and STEEP for the three simulation scenarios.

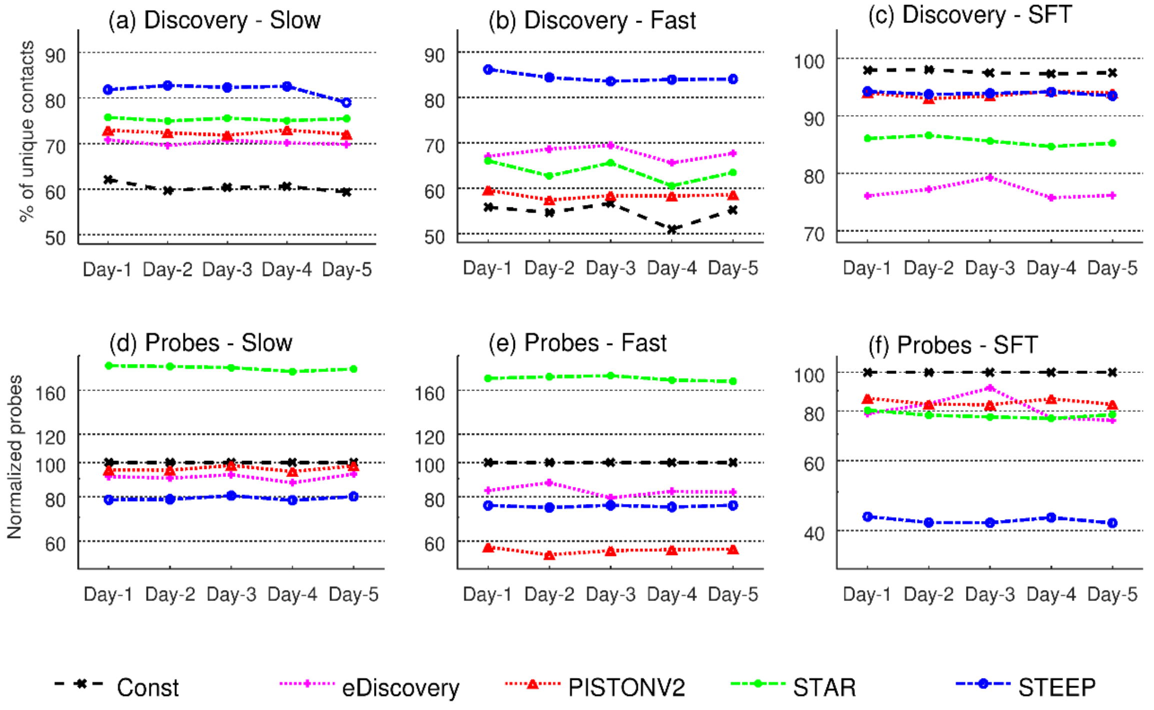


Figure 6.6 Average performances of algorithms over several days across scenarios

Regarding the unique discoveries per day, STEEP performs best in terms of slow and fast, while it is nearly the best for the SFT scenario. Similarly, for probes, STEEP conserves the maximum amount of energy for the slow and SFT scenarios. In the slow scenario, STEEP conserves at least 10% more energy than the closest alternative (eDiscovery) for probes in Figure 6.6(d), while it has the best performance regarding unique contacts in corresponding Figure 6.6(a). STAR is second best for unique contacts but consumes almost twice the energy for probes. In the fast scenario, Figure 6.6(b) shows that STEEP is the best for unique contacts by a significant margin (15-20%) relative to the second-best algorithm (eDiscovery). Correlating this with Figure 6.6(e) regarding energy consumption, only PISTONv2 conserves more energy that STEEP. The relative 20% power savings of PISTONv2 comes at the cost of a 25% loss in unique contacts. For the SFT scenario, Const has the best performance regarding unique contacts (Figure 6.6(c)). STEEP and PISTONv2 are almost tied regarding unique contacts and are within 5% of Const. STEEP consumes almost half the energy compared to others as shown in Figure 6.6(f).

As discussed in the context of Table 6.6, among the three simulation scenarios, the SFT nodes are mobile most of the time. This is the primary reason why the algorithm metrics had to be varied as shown in Table 6.7. Only STEEP and PISTONv2 did not require constants to be varied, as they include speed within their IPT computations. The relation between STEEP and PISTONv2 is not as prominent in the slow and fast scenarios, as the nodes are static most of the time. In the SFT scenario, for probe counts, there is a clear correlation between PISTONv2 and STEEP. When compared to PISTONv2, the probe counts for STEEP are almost half in Figure 6.6(f). This is because AF_{disc} is set to 1 for STEEP (Eq. 6.4), while the corresponding value for PISTONv2 is 2 (Eq. 6.1). Despite this change in the IPT computation, STEEP does not degrade in terms of discovery because of the LSE logic. It detects sudden speed changes and responds by reducing the IPT (Algo 6.3), thus improving the discovery. It can be observed that the contribution to probes from the last significant event time is very small (less than 10% in the case of SFTs).

The probe count results for STEEP and PISTONv2 appear to contradict the above discussion in the case of the fast simulation runs of Figure 6.6(e). An in-depth analysis with respect to Eq. 6.15 shows that a high value of M increases the probes for STEEP. In the case of WDM-fast, the nodes either stay static or move using cars/buses. This condition results in a large value for M. Since the nodes in WDM-slow are static and have a mix of pedestrian- and bus-based mobility across the speed changes, the values for M are lower. Similarly, cars in the case of real-world traces move at different speeds and move most of the time. Thus, the value of M is relatively unimportant. An examination of the simulation traces supports this analysis. When moving from stationary to car/bus speeds (in the fast scenario), STEEP sends significantly more probes after Algo 6.3 detects the change in speed and reduces the IPT. For example, if the IPT is computed before a person starts traveling from home to work, then the IPT is computed as MaxIPT. If the drive to the office is only a few minutes long, PISTONv2 will not send any probes during travel. If the next IPT is computed after reaching the office, the speed may once again be slow, causing the IPT to remain high. On the other hand, STEEP will notice the change in speed within Speed_ChkIntrvl (60 seconds in this case) after starting the trip to the office. Since cars drive at significantly large speeds, STEEP will compute low values for the IPT while traveling to the office, thus increasing the number of probes. These additional probes are responsible for the significantly better discovery.

For WDM-slow, nodes require a long time to reach their destination, as they travel at lower speeds while walking or take buses that need not travel via the shortest path. However, STEEP notices the change in speed much earlier than PISTONv2, since the speed is low, the IPT values remain at moderate levels. In most cases, the travel duration for the slow scenario is greater than MaxIPT. This causes PISTONv2 to start sending probes during travel. As discussed above, for these experiments, PISTONv2 will probe at least twice as aggressively as STEEP as α is set to one. Though PISTONv2 may not send any probes during the initial part of movement start (up to MaxIPT), it subsequently sends many probes, resulting in poorer performance than that of STEEP for Figure 6.6(d). Since STEEP starts probing earlier, it discovers more unique contacts, as shown in Figure 6.6(a).

6.6.3. *Contact duration spread and the total number of contacts*

Figure 6.7 plots the spread in contact duration for all runs. It does not differentiate the uniqueness of contacts and only captures the contacts and their duration. Table 6.9 presents the cumulative number of contacts for different algorithms across the simulation runs. In essence, the values in Table 6.9 represent the area under the curve for various algorithms.

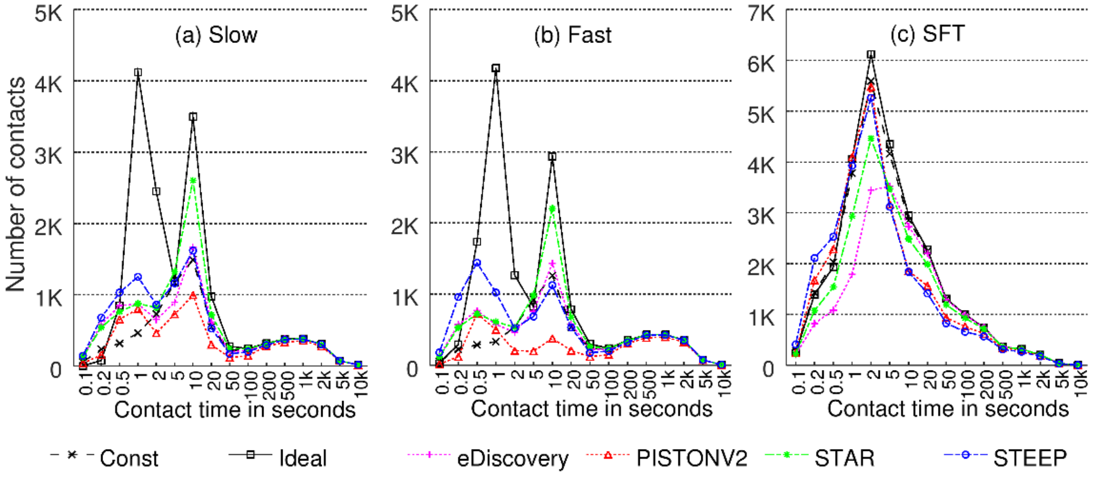


Figure 6.7 Contact duration spread for simulation across scenarios

Disregarding the hypothetical ideal case, STAR reports the best values for the slow and fast scenarios, while Const has the best value for the SFT scenario. This is expected because the number of probes used by STAR in the slow and fast scenarios (see Figure 6.6(d) and Figure 6.6(e)) is quite high. Similarly, in Figure 6.6(f), Const sends the maximum number of probes; hence, it is expected to discover more contacts on a cumulative basis. An interesting observation for STAR is that while it discovers more contacts on a cumulative basis, it performs poorly in terms of diversity (i.e., daily unique contacts). This can be attributed to the fact that the use of historical data tends to bias the discovery of nodes that belong to the same community. Since PISTONv2 uses speed and STEEP borrows this element, they both make discoveries with higher diversity. The benefit of diversity is also visible in the routing results (discussed below).

Table 6.9 Total number of contacts across discovery algorithms

Discovery Approach	**Slow**	**Fast**	**SFT**
Ideal	15175	14252	27378
Const	6956	6257	26478
eDiscovery	8216	7400	19839
PISTONv2	5761	4097	23647
STAR	9806	8562	22036
STEEP	9161	8509	23620

As shown in Figure 6.7(a) and Figure 6.7(b), in the ideal scenario with continuous polling, no contacts are missed, and the largest area under the plot curve is achieved. As we move from the ideal scenario to discrete probes, two things are observed: a) the area under the curve for the discrete protocols decreases because some contacts are missed; b) the plots for these algorithms shift towards the left as the contacts are reported for shorter durations (since the probe is sent later).

Under the ideal scenario, WDM-slow has approximately 5,000 contacts of one second or less and another 7,000 contacts between one and ten seconds. In contrast, WDM-fast has approximately 6,200 contacts of one second or less and only 5,000 contacts between one and ten seconds. Beyond ten seconds, the plots are almost identical. Hence, WDM-fast (Figure 6.6(b)) will yield a greater number of missed contacts compared to WDM-slow (Figure 6.6(a)).

Compared to WDM scenarios in which we see two peaks at 1 second and 10 seconds, with a drop in between, the SFT scenario has a more continuous spread, with a higher peak at 2 seconds and much larger values for 5 seconds. The total number of contacts for the SFT scenario is also much higher, as shown in Table 6.9.

6.6.4. *Impact of discovery algorithms on routing*

Figure 6.8 illustrates the impact of different discovery algorithms on spray-and-wait-based DTN routing. As expected, the ideal discovery performs the best in the delivery of DTN bundles. STEEP is a clear winner among the adaptive algorithms for the slow and fast scenarios. In the case of SFTs, almost all adaptive algorithms except eDiscovery have very similar performances. As discussed w.r.t. Figure 6.6(f), after conserving 40% to 60% of the energy spent on probes, STEEP performs well, even when routing under various conditions.

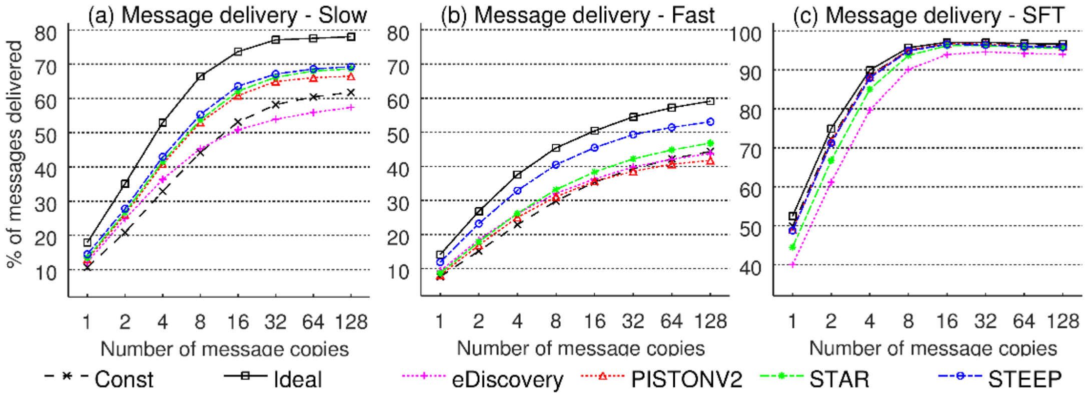


Figure 6.8 Performances across scenarios for SNW for different numbers of copies

Comparing Figure 6.8(b) with Figure 6.6(b) and Table 6.9 reveals that the delivery ratio correlates with the number of unique contacts but not with the total number of contacts. For example, STAR performs better regarding total discovery in the fast scenario (see Table 6.9) but performs significantly poorer than STEEP in terms of routing.

6.6.5. *Performances for different node densities*

The goal of these experiments is to study the adaptability of the discovery algorithms for various network densities. The SFT scenario is not included in these runs because the taxi nodes demonstrated a very large number of contacts in the taxi stand when the number of taxis was increased to two hundred or more. For both scenarios, when we consider the trade-off between unique contacts and number of probes, STEEP performs the best.

Figure 6.9(a) and Figure 6.9(c) display the density-based results for normalized unique contacts and normalized probe counts in the slow scenario. The discovery rate decreases for all algorithms as the density increases from 25 nodes to 100 nodes. Const, PISTONv2, and STEEP do not show an improvement in normalized unique contacts as the number of nodes further increases. This is because the IPT is fixed for Const, while PISTONv2 and STEEP primarily depend on the node speed. STAR and eDiscovery show an improvement in unique contacts for the slow scenario as the number of nodes increases beyond 200. The logarithmic plot of Figure 6.9(c) shows that the improvement comes at the cost of an increase in the number of probes for eDiscovery and STAR. As the density increases, STAR discovers more nodes, and its estimate of contacts (λ in Eq. 6.2) becomes higher, leading to a smaller IPT value. Similarly, eDiscovery discovers frequent contacts, causing its IPT values to be closer to base_I. The relative increase in the number of probes for STAR and eDiscovery is significantly higher than the improvement in unique contacts. In fact, at higher densities, STEEP misses approximately 5% of the contacts when compared to STAR and eDiscovery, while conserving 50% or more energy regarding normalized probes.

For the fast scenario, in Figure 6.9(b) the number of unique contacts decreases for all algorithms except STAR. STEEP performs the best at all densities. However, STAR and eDiscovery exhibit an increase in the number of probes (Figure 6.9(d)) with density, their discovery efficiency is nowhere near that of STEEP. Similar to Figure 6.6(e), Figure 6.9(d) shows that PISTONv2 conserves the most energy regarding probes, but it degrades worse than STEEP in terms of the number of unique encounters. The unique encounter gap between STEEP and PISTONv2 increases from 20% at low density to 30% at high density, while the percentage of energy conserved regarding probes remains at approximately 20%.

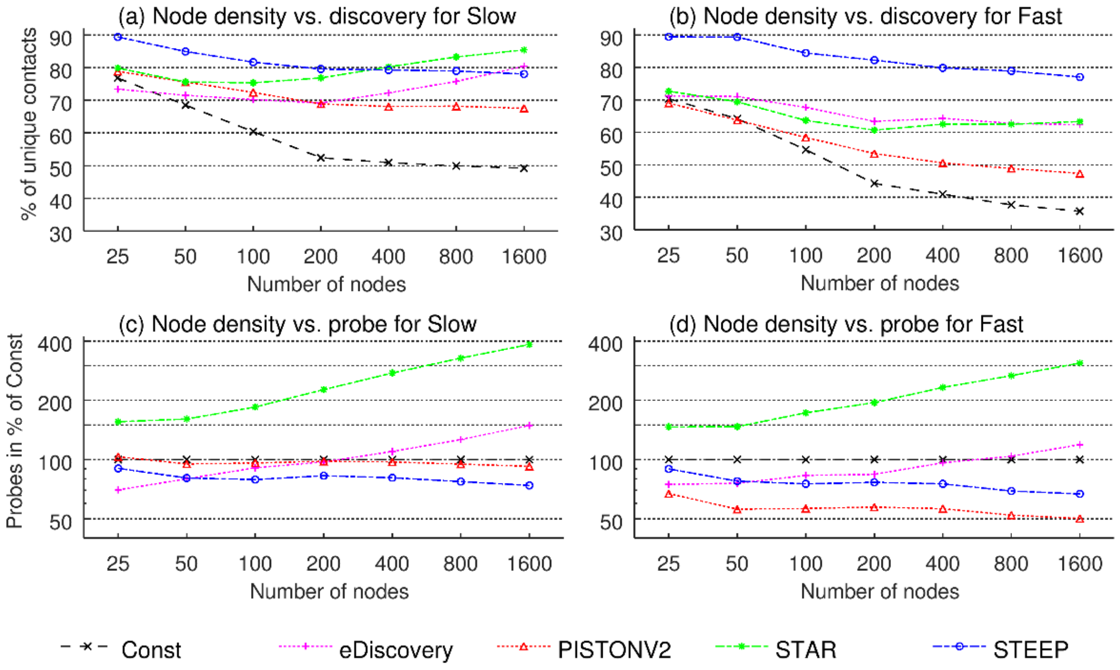


Figure 6.9 Performances of discovery algorithms in WDM for different node densities

6.6.6. *Impact of energy saving measures on the discovery*

Figure 6.10(a) and Figure 6.10(b) present the number of unique encounters per day as nodes attempt to conserve energy by entering sleep states between probe transmissions. The target SP is varied in multiples of 10 from 0% to 60%. The probe count is shown in Figure 6.10(c) and Figure 6.10(d). Since constant probing would synchronize the sleep states, it was excluded from this experiment. The STAR discovery algorithm wakes up the interface for probing much more frequently than other algorithms. Thus, the interface slept for a shorter percentage of the time relative to the target. The other discovery algorithms were within 90% of the sleep target.

PISTONv2 exhibits a flat line for probes, as its IPT computation does not consider the contact history. STAR and eDiscovery rely on historical contacts. Hence, as the nodes sleep for longer intervals, they may not respond to probes from other nodes, leading to discovery failure. The cumulative effect would cause both these algorithms to increase their IPTs, leading to further decreases in discovery. Hence, for STAR and eDiscovery, the slope for probes is sharper than a linear drop.

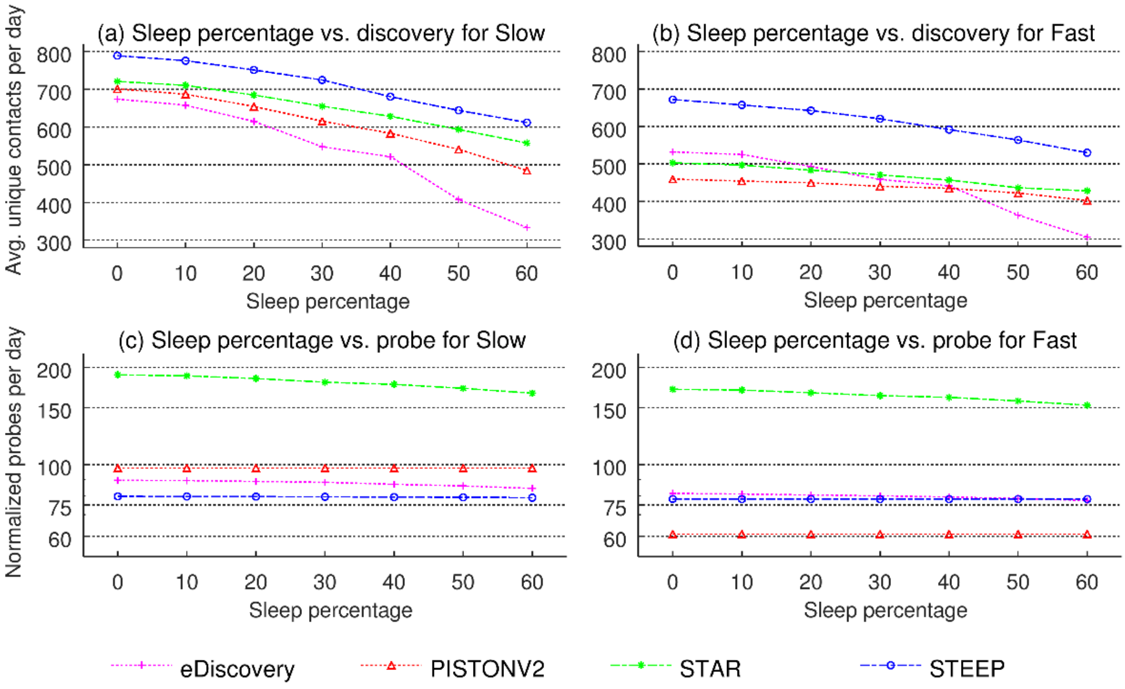


Figure 6.10 Performances of discovery algorithms in WDM for different sleep targets

Regarding the amount of energy spent on probes, STEEP consistently had the lowest performance, while it performed well regarding the discovery of unique contacts. In fact, if only discoveries and short message communication are targeted, STEEP works reasonably well compared to the other algorithms across different sleep targets.

Based on the simulation results for our selected initial energy, the average node lifetime was approximately 5.9 days for no sleep and 14.5 days for a 60% sleep target. For WDM-fast, the discovery performance of STEEP with 50-60% sleep was as good as that of other algorithms without sleep. Similarly, for WDM-slow, STEEP performed better regarding discovery than the second-best algorithm, even with an extra 30% sleep interval. This shows that in the absence of a significant communication payload, STEEP can more than double the battery life while retaining the discovery percentage for the fast scenario. For the slow scenario, the increase in battery life was in the range of 40-50%.

6.6.7. *Evaluation of different components of STEEP in the base scenario*

Figure 6.11 captures the performance of STEEP when varying one of the algorithmic constants at a time while keeping all others at their default values (see Table 6.8). Interestingly, the plot for real-world traces (SFT) shows a uniform curve in all subplots, while the WDM simulations react significantly to changes across the runs. This can be attributed to the fact that SFT nodes are mobile most of the time and have a wider distribution of contact durations, as shown in Figure 6.5(a).

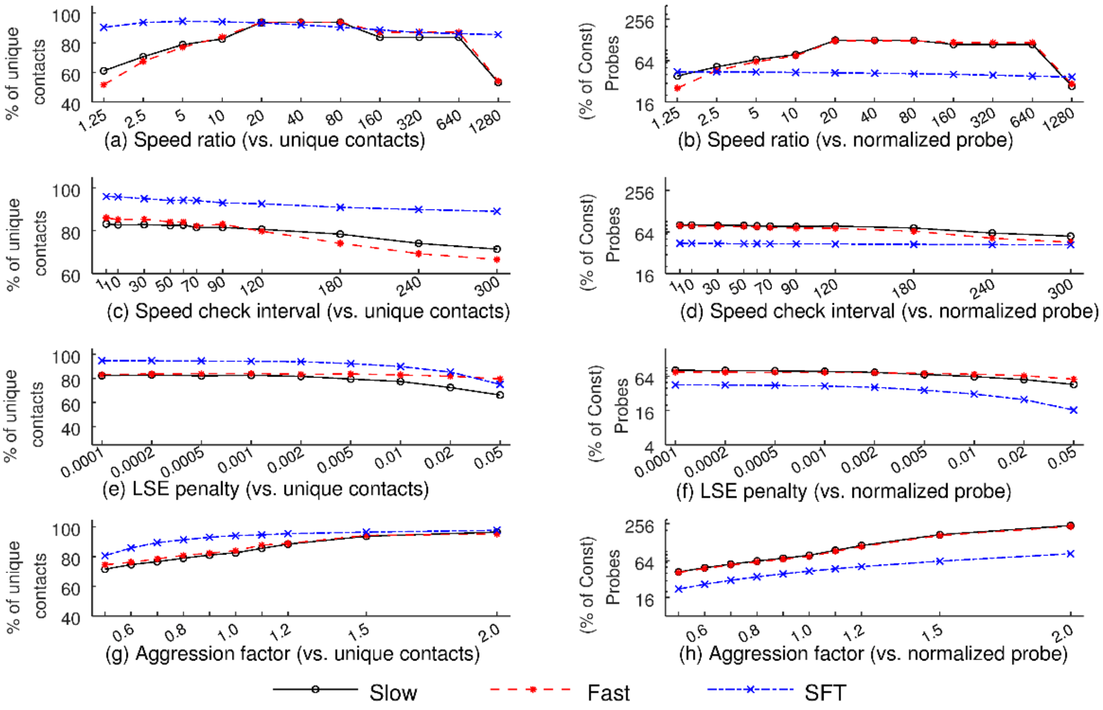


Figure 6.11 Performances of STEEP for different parameters – default values

For the slow and fast scenarios, in Figure 6.11(a and b), it can be observed that the number of unique contacts is not impacted as we increase the speed ratio from 20 to 80. Choosing very high values (e.g., 160) as well as low values (below 5 seconds) does reduce the discovery of unique contacts. Similarly, in Figure 6.11(c and d), for the speed check interval, the efficiency in terms of unique contacts drops as the check interval increases beyond 90 seconds. It should be noted that our selection of a default value of 60 seconds for the check interval and 10 for the speed ratio was made to align with real-world numbers rather than to fit the values with the best possible results. In fact, the range of values that could be chosen is quite wide across these three scenarios.

The impact of $\text{LSE}_{\text{penalty}}$ across scenarios is plotted in Figure 6.11(e and f). The number of unique contacts begins to decrease as we increase the penalty beyond 0.002. Figure 6.11(g and h) show that as the aggression factor increases, the number of probes increases linearly on a logarithmic scale, while the increase in the number of unique contacts flattens off. For the SFT scenario, we do not see significant benefits regarding the number of unique contacts beyond AF = 1.2, though we continue to increase the number of probes as the value increases. Once again, the choice of 0.01 for $\text{LSE}_{\text{penalty}}$ and 1.0 for AF is not an attempt to over-fit the algorithmic constants to the observed results.

6.6.8. *Evaluation of different components of STEEP in isolation*

Figure 6.12 captures the performance of STEEP as we vary the values of the speed factor, check interval, $\text{LSE}_{\text{penalty}}$ and aggression factor to evaluate the impact of each of the components in isolation. Figure 6.12(a and d) analyzes the speed check algorithm with the $\text{LSE}_{\text{penalty}}$ set to 0 and the aggression factor set to 2 (similar to PISTONv2). Since the penalty is set to zero, the discovery rates are somewhat better, but the number of probes increases significantly when compared with Figure 6.11(b and d). For speed ratios below 5, some of the contacts are missed in the WDM scenarios, but the rate of discovery remains high until reaching a speed ratio of 80. Regarding the check interval variation, the WDM simulations show a decrease in discovery values (associated with the reduction in energy spent on probes) as we vary the check interval above 50 seconds.

Variation of the $\text{LSE}_{\text{penalty}}$ in Figure 6.12(e) and 12(f), and variation of aggression factor in Figure 6.12(g and h), yield similar trends, as shown in Figure 6.11, except that the discovery rates and numbers of probes have much lower values. The $\text{LSE}_{\text{penalty}}$ plots are for the scenario in which the speed ratio is set very high; hence, incremental improvements occur only with respect to the OnConnectionUp logic, as shown in Algo 6.4.

As the $LSE_{penalty}$ value increases beyond 0.002, the number of unique contacts decreases. For the AF plots, $LSE_{penalty}$ is set to 0. This makes Eq. 6.4 (for STEEP) similar to Eq. 6.1 (for PISTONv2), with AF_{disc} replacing 2α. As expected, the results for this isolated run follow the trends of PISTONv2. As AF decreases to values below 0.8, the number of unique contacts becomes significantly lower. This corresponds to the observation in [72], where α values below 0.3 resulted in several missed contacts.

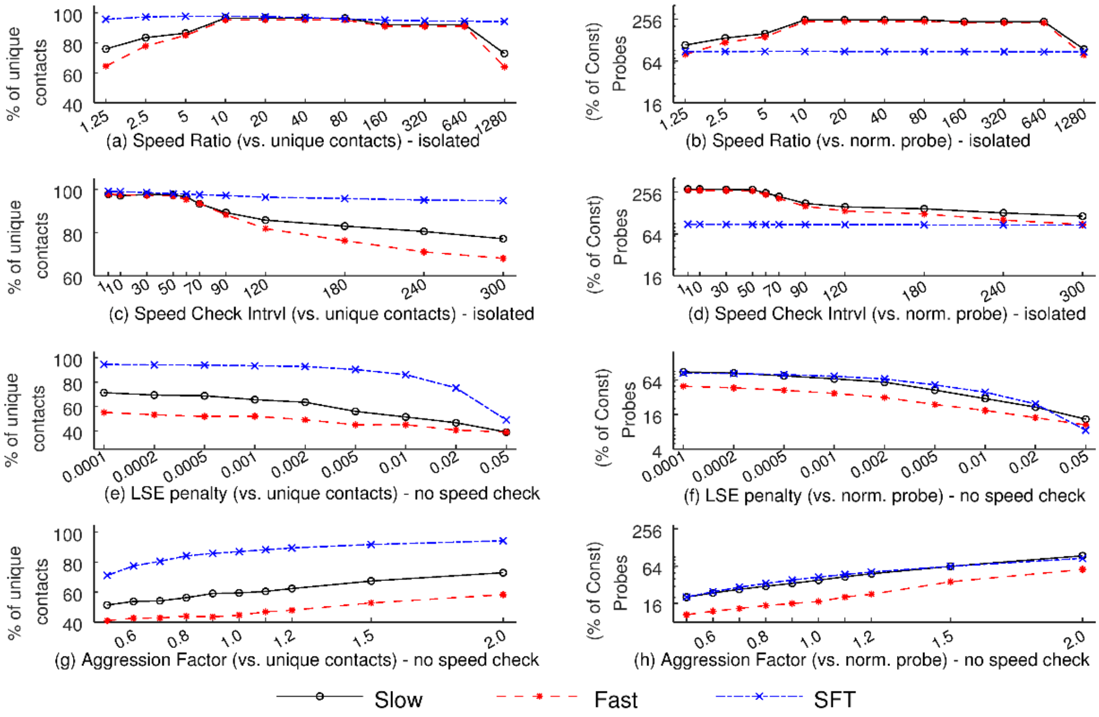


Figure 6.12 Performances of STEEP for different parameters (isolated contribution)

Based on a combined evaluation of Figure 6.11 and Figure 6.12, it can be observed that the SFT scenario provides much larger ranges for all constants (e.g., from Figure 6.11(a,b), Figure 6.12(a,b), the speed ratio can be between 2.5 and 80 for the SFT scenario). The slow and fast scenarios have slightly smaller ranges (e.g., the speed ratio can be varied between 10 and 80) for similar performances. The same set of default values of STEEP (as shown in Table 6.8) provides good results irrespective of the mobility scenarios and node densities. Noting that the chosen values were not the best results from Figure 6.11 and Figure 6.12, we can infer that STEEP may not need to adapt its algorithmic constants to different scenarios.

6.7. Summary

We have presented STEEP, an adaptive energy-efficient device discovery protocol for opportunistic networks based on the node speed and information regarding the last contact. The simulation results show that in most scenarios, STEEP performs better in terms of discovery while consuming 20-30% less energy to transmit probes relative to other algorithms.

We also analyzed the scalability of discovery algorithms for various node densities. STEEP demonstrates satisfactory scalability relative to other approaches. STEEP achieves an excellent discovery performance for both high and low densities; as the density increases, the probes per node are kept constant. Furthermore, we experimented with turning off devices to conserve power. The simulation results showed that STEEP could provide discovery rates similar to that of other algorithms (eDiscovery, STAR, and PISTONv2) even if the nodes sleep for 30-50% longer durations. Thus, for sparse networks with low communication loads, nodes can significantly extend their battery lives using STEEP.

STEEP outperforms other adaptive algorithms for DTN routing runs (spray-and-wait with varying copy counts) while conserving power when transmitting probes. The results demonstrate that the number of unique contacts is more important for routing than the total number of contacts. Furthermore, it was observed that STEEP does not require constants to be adapted to different deployments.

In the future, we aim to analyze the impacts of these algorithms in other real-world scenarios, particularly those involving multiple nodes with a mix of pedestrian and vehicular mobility. A possible future extension is to implement STEEP on real devices.

7. DEPLOYING OPP-NETS

Summary—In this chapter, we explore a couple of deployments of Opp-Nets in the real world. We attempt to create a rural environment within our institute to communicate over these platforms. In the first sub-section, we study the requirements of deployments in rural areas to be sustainable in the long run. Threafter, we study the design and deployment of SWiFiIC, a platform that supports options to generate revenue at the rural community level. We also demonstrate a couple of end-user applications (text and image sharing) on top of this platform. SWiFiIC uses an approach that is similar to QoseCo-CR, to route all messages via a central Hub using DTN stack running on top of WiFi. Its key design objective is to make it self-sustainable by having revenue generation abilities, and extensible to newer applications. Further, it targets to be easy to manage the user lifecycle, while having provisions for compliance with local policies.

In the third sub-section, we study the design and deployment of Vectors. This is a distributed Opp-Net on top of Android Nearby to show-case communication of SVC video chunks. It showcases running of a simplified version of SORT as covered in 3 over multiple days within the campus.

7.1. Making rural deployments sustainable

Internet usage in rural areas of most developing nations has not been encouraging. Telecom operators have found it commercially unviable to offer Internet services in rural areas because of low revenue and scarcity of electrical power. Small Internet service providers have not succeeded because the existing solutions have been complex to install and operate, leading to increased costs. We present an overview of the challenges for rural connectivity and review some of the existing solutions in improving rural connectivity.

Developing countries like India are struggling to provide good quality of living in rural areas. This is especially evident in the lack of employment opportunities, lack of education and lack of medical facilities. It was expected that Internet would bridge this divide but, broadband failed to penetrate rural areas. As discussed in 1, even developed countries find it unviable to provide good broadband connectivity to rural areas.

Proactive initiatives focused on creating a fund for rural connectivity has not been able to make a significant impact using telecom infrastructure. Mobile connectivity for voice has improved significantly over the last decade in India, but this has not improved broadband penetration. Relatively higher cost of user terminal (devices), high cost of bandwidth and lack of applications for rural users have discouraged wider deployments. Low population density and less disposable income discourages telecom operators from making further investments. Moreover, telecom operators have had the burden of additional operational expenses because of frequent power cuts. Many of the rural areas have also started providing Internet connectivity to mobile phones. But the actual speeds in many rural areas are nowhere close to what can be considered broadband speeds [1-3].

With smart devices based on Android and similar platforms, user terminal device costs have significantly reduced and are expected to be affordable for masses with operator or government subsidy. For example, telecom service providers sell devices with subsidy as long as users commit to spending a minimum amount for their service.

This provides a significant opportunity for affordable broadband connectivity. Large sized telecom operators are not able to offer Internet access at price points that make its usage realistic for the majority of rural population. Government initiatives can drive the costs down using subsidies, but lack of self-sustainable solutions will imply that once the subsidy is removed, these services become operationally unviable.

In developed countries (and even in India for areas which have a high demand for network), next-generation mobile wireless technologies and hot-spots based on wireless access points (AP) have been deployed and are operating beneficially. Multiple deployments have also been carried out in rural Europe for community operated wireless networks. But such solutions do not scale down and scale out for areas with sparse population, with low expendable income and low literacy. The primary problem with scaling down and scaling out is tied to technical expertise needed to operate such specialized networks. Moreover, existing applications using these networks in urban scenarios do not have the same relevance in rural scenarios because of differences in needs and lack of education.

Majority of the productive applications presently being enabled in rural community are pushing information from Internet towards rural community for targeted domains like weather, agriculture, transport, health etc.[88]. These apps do not have any incentives from the consumer for the application developer, and generally need subsidies to operate in the long run.

7.1.1. *Motivation and overview of prior work for rural connectivity*

A lot of research has been done in last fifteen years for rural connectivity. The research around last hop connectivity can be categorized into six broad areas, namely , a) extension of wired networks to rural areas; b) satellite-based networks; c) extension of telecom infrastructure based (LTE/WiMAX) wireless networks to rural areas; d) wireless mesh networks e) 802.11 access point based networks with wired/wireless connectivity towards the core; and f) solutions utilizing DTN / Opp-Nets.

The cost of laying fiber or copper in rural areas for end user connectivity is high. Optical access technologies like PON based fiber to the home scenarios have not been considered because of cost of device and deployment cost, besides management complexity. Fiber to the curb (FTTC) solution can be the backbone connectivity for providing rural connectivity (e.g., BharatNet or NOFN [89] connecting the Gram Panchayats). But FTTC does not solve the last hop connectivity. Further, the operational expenses for servicing these setups continue to be significantly high.

Similarly, scenarios involving extension of wired networks in rural area perform worse than fiber because of additional electrical power demands. This tends to increase both capital and operational expenses. Moreover, the energy and maintenance demands for rural areas are not financially viable. S Bhaumik et al. in [90] have proposed fiber deep and selective placements of DSLAMs to minimize power costs, and still they cannot scale down to the needs of rural areas of developing countries.

Satellite-based networks provide most of the benefits while continuing to work with frequent power outages etc. The most significant issue with usage of satellite as access network in rural areas is related to the cost per user. Satellite communication induces higher latency and hinders two-way communication. An interesting approach was considered in [91] wherein both satellite and GPRS based communication were brought together. Such approach adds to the complexity of network regarding management and deployment but allows predictable degradation to service. They have solved the challenges of configuration corruption and cost of reconfiguration etc. using a simple backup for important files across the systems. The authors in [91] have looked at management aspects using Nagios for alerting against failures.

Note that hybrid approaches like Google's Loon project [92], Facebook's drone-based initiatives [93], etc. may bring more affordable Internet in future. Such approaches and fiber, copper or satellite-based communication can help connect rural platforms to the core network.

a) *Wireless mesh network (Rural Wireless Network Operators)*

Amongst the rural connectivity approaches, wireless mesh networks have proven very cost-effective to deploy. High-performance WiFi systems allow line of sight deployments that may span tens of kilometers [94]. Emerging technologies like LiFi [95] will help provide even higher bandwidth.

Patra[94] and Backens [96], highlight some of the reasons why other alternatives have not flourished in rural areas, such as, unavailability of expertise; lack of reliable power; capital-intensive business models, etc. They also highlight the benefit of using wireless technologies in rural areas– i.e., cleaner spectrum because of fewer number of RF transmitting device. Similar projects in India in the past have included Digital Gangetic Plains project[97], Fractal project, AirJaldi, etc.

We analyzed these deployments and found that mesh-based solutions bring in network connectivity, but the ownership is not transferred to local communities. The technology and operational aspects stay with the research team, and though local people know how to use the system, they cannot maintain, upgrade or operate the system independently.

Mesh based deployments have multi-hop latencies and may limit two-way communications. Moreover, if the solution does not use commodity hardware, it increases technical complexities making it costlier.

AirJaldi is a semi-urban wireless network operator [98] that has a long-running and sustainable operations in semi-rural areas primarily serving semi-rural enterprises. This is a niche that they can serve by deploying line of sight long distance wireless communication protocols. Their customers have close to enterprise type needs, which provides them good revenue stream, and hence they can afford trained/hired in-house experts to run the systems. They have adapted themselves to technical challenges of operating a network service in sparsely populated areas where both power supply constraints and environmental issues like loss of RF because of rain etc. are a concern. While AirJaldi can offer services in semi-urban scenarios, most of their deployments are in areas with high population density.

b) *Telecom networks*

With time, data speeds over telecom cellular networks have increased, and now even rural areas can offer 3G/4G connectivity at speeds up to megabits per second. Despite this, the revenue model for telecom networks is not sufficient to support rural data needs. Report by PWC[99] suggests that urban to rural acquisition cost of customer is 50 % higher while servicing cost is 25% higher, whereas ARPU is 20% lower.

Overall the telecom service providers have their interests in profits and are not agile enough to cater to local community needs, especially faced with multiple local languages, low literacy, etc.

c) *802.11 Access point based networks*

Access points have frequently been deployed in urban areas, like cafes, transport transit points, and hotels. In the urban scenarios, the facility owner lacks in-house expertise, and hence they outsource facility/network service management. The firms that provide operations management and billing bring in their specialized hardware and software to minimize their operational expense.

The same service providers will not be suitable for rural areas primarily because of cost of operations (time to travel, service, etc.) and the fact that target end-users are not as tech savvy as urban users of these solutions. Many times, the custom hardware is unsuitable to power fluctuations, dust, and temperatures in rural areas. One interesting solution to this problem-area has been successfully deployed as “Fon Global WiFi Network” where the end-customer crowd-source to share spare broadband bandwidth on a best effort basis. As of April 2018 Fon claims to have 20 million residential hotspots, 310 Airports, 24,000 hotels and 850,000 other venues covered by their access points, primarily in Europe [100].

d) *Contrast with cable TV operators*

We tried to compare Internet penetration with Cable TV penetration in India because while Internet struggles to find a foothold in rural areas, Cable TV has been quite successful. This disparity can be explained in:

a. Degraded video is acceptable; degraded Internet service may not be a acceptable scenario for many applications.

b. Cable TV was using analog signals; network management and maintenance are similar to what an electrical technician can learn in a few days. Gradually they have been trained for digital services.

c. Video has a lot of local language content. Many TV channels provide content curated for a city or state.

d. Video does not need a return path, while Internet relies on interactivity and return path for many applications.

Compared with cable service, it is important to note that DOCSIS based Internet connectivity on top of cable infrastructure has not been successful in India. The two-way nature of Internet access needs return path, which adds to the complexity of network, thus increasing operational costs significantly.

Local cable operators employ a significant number of people who are touch points for the end customers. If an efficient platform is created for rural Internet connectivity, employment opportunity will be of the same order as that of cable services or possibly higher.

e) *Delay tolerant networks for rural connectivity*

For deployments within communities, TierStore [101] was one of the early attempts at content sharing. Some of the other implementations are MotoPost[102] for voice messaging, and DakNet[103] for multiple scenarios including information retrieval by relying on vehicular ad-hoc networks. Further applications have been developed using stacks available in open source like IBR-DTN [104] and other options like DTN2, JDTN, etc.[105]. DakNet [103] and other research [98] have highlighted benefits that exist even in non-real time communication over disconnected networks using vehicular MANETs. Delay tolerant network architecture [106] allows much more flexibility for such scenarios. Research labs have produced SDK for DTN, but eco-systems using DTN have not been widely deployed in an open/extensible manner. Moreover, almost all the research work has ignored monetization of such service. Demmer [12] in his research work has suggested a publisher-subscriber-based design that can be optimally used to provide a platform for application development. The Internet Village Motoman [108] project analyzed deployment and its end usage behaviors. It found some unintended usage , besides lack of communication for intentional usage scenarios.

DakNet combined the benefits of mesh networks with delay tolerant networks using buses as ferry for data transfer. The solution in [103] was able to reach deployment cost of below $200 per service center. The use case chosen for optimizing DakNet was land records in rural areas near Bangalore (Bhoomi project). Though DakNet-Bhoomi offered an interesting use case for local communities, the framework did not provide a platform that fostered further application development.

Lindgren et al. in [109] have highlighted the fact that DTN has not succeeded in scenarios other than deep space exploration[110]. By bringing in a revenue angle to DTN and crowdsourcing delivery of bundles we can provide a solution that possibly fulfills real needs. Another key learning for rural deployments is to have a platform on which newer apps can be developed based on user needs. Applications designed by urban people for rural needs may not be compatible to solve the problems or may be sub-optimal in serving their needs.

7.1.2. *Ideal characteristics for last hop rural connectivity:*

Having analyzed the existing solutions, it becomes apparent that pure Internet connectivity in rural scenarios will not provide a good user experience unless the quality of service is maintained. The cost of having regular electrical power and backend connectivity is significant. Hence the local rural service provider will need to have a solution that handles both electrical power problems and bandwidth problems gracefully.

As the study above shows, none of the deployed solutions provided an eco-system for community to engage locally. They only ensured that the data from Internet was made available to them. While some of those data scenarios (like land record, weather, etc.) make sense, it is difficult for end user to access them as they are. Thus, they tend to rely on Kiosk-like systems where information is adapted to suit their consumption. Sometimes, the adaptation may be made manually. Ideally, a rural Internet infrastructure platform should provide automated adaption of data on Internet to a form that end user can easily understand and consume from their device, rather than having to come to a kiosk. Since this cannot be done generically in advance, the authors propose a framework in which applications can be quickly developed to do such transformations while minimizing bandwidth needs, etc.

Based on the above review and learning from [111], the main requirements for a system that would possibly succeed to sustain itself once deployed, are identified as:

1) Infrastructure hardware operable in rural environment, –tolerant to extreme temperatures, dust and moisture; with shielding for shade
2) Possibility of using commodity hardware to maximum, which should lower servicing costs and management cost, besides scope for easily training people.
3) Ensuring ease of network maintenance by the local operator, which should reduce needs for experts, or need travel to city for resolution of problems.

4) Keeping the cost of user management minimum including user addition (identity, address proof), and billing (usage report, recharge).
5) Ensuring security and regulatory compliances with minimum end user impact.
6) Providing an eco-system that engages local community enabling localized application scenarios, with easy access to local content.
7) Providing graceful degradation of service when power is lost or back-end network connectivity is lost.

While the above list includes integrated connectivity to Internet, the focus of our thesis is primarily communication within the village. The next section covers design and deployment details for SWiFiIC keeping in mind some of the properties identified above.

7.2. System I - SWiFiIC

In this sub-section, we discuss a SWiFiIC, deployed using open source software and commodity hardware, customized to simplify deployment, management and operations of WiFi-based DTN. SWiFiIC is an effort to create an extensible platform for future applications in rural communication where delay and loss are less significant when compared to cost.

The most frequent management scenarios around customer acquisition, billing/recharge, and security are simplified using an integrated application. The system targets small-scale rural operators who deploy extensible apps for end users, generating revenue while keeping it affordable. To the best of our knowledge, an extensible platform targeting an eco-system for rural digital communication does not exist.

To ensure sustainability, it is necessary to have engagement and revenue generation from within the local community. Our proposed system allows lightweight apps to be deployed at the community level, such that they enable revenue opportunities for the platform provider as well as the end users. The central component of this system is a simple computer-based platform called AppHub (also referred to as hub in the subsequent section). Local applications have server components deployed on AppHub, and these components utilize infrastructure from the operator (compute, storage, end-user identity, billing, and network). Since most of these apps target local communication, the Hub will allow them to function even when the connectivity to the core of Internet is not present. In a way, AppHub operates similarly to the central hub of QoseCo-CR, discussed in 5. We assume that these apps are delay tolerant by default and may get close to real-time performance when nodes are in the vicinity of hub.

The demography of users in rural areas does not demand Internet connectivity for the majority of their productive communication needs; hence we can use DTNs or Opp-Nets in these scenarios. The platform allows quick development of localized applications for day to day needs. Using data-driven paradigm, the solution simplifies and abstracts clients and server implementations for applications. For more intensive applications, the platform allows modular expansion on both client and server pieces of applications.

Certain parts of the system will rely on Internet connectivity (e.g., firmware upgrade, licensing, etc.), and they will be implemented using modularly developed customized solutions. The current work assumes that authenticity and privacy (if desired), will utilize solutions similar to those in public key infrastructure. The proposed architecture allows such a deployment. Attempts to bring in security as part of initial design caused a negative impact on usability. Since security complicates the deployment, security and trust implications have been deferred to as future work.

To prove extensibility and operational efficiency, we present the design and implementation details of two end-user apps, targeting text messaging between users and image sharing on top of the platform. In the past, we had also implemented prototype applications for conducting online evaluations for school tests and grocery supply from local grocery shops.

Target end users for SWiFiIC Apps are rural people with limited tech-savviness. Most apps are expected to be purely graphical so that even illiterate people can use them.

The implementation of the SWiFiIC project involves multiple applications for end-user usage. For this study, only a subset of applications is included to complement the claims of contributions made in this chapter in particular, and the thesis in general.

- SWiFiIC Operator App (SOApp) - Use case for User addition: to showcase the claim of simplicity in operations and management.
- SWiFiIC User Terminal App (SUTA) – to showcase user experience on billing, new application download, etc.
- Messaging and photo sharing app – to showcase usage of mixed media content.

Deployment of applications developed from urban perspective has had limited success. They have been either a) complex for users or b) complex to deploy and operate or c) lacked genuine need. In the context of present technical barriers, rural folks are not positioned to understand and express applications for their need. Observations in the village scenario bring out multiple aspects where digital communication can greatly improve productivity. We also understand that we are not the right persons to dictate the apps to be developed. Hence, we focus more on the SWiFiIC platform and its extensibility.

7.2.1. *Design choices*

The choice of end-user device platform in wireless realm logically led us to Android, the most widely deployed platform which is expected to cross the price barrier ensuring affordability for rural masses.

While some of the applications are stand-alone, the target of connecting rural folks implies that the applications need to communicate with each other or with a central entity. We analyzed P2P based applications and found them too complex to deploy and manage in rural environments. On the other hand, server-based infrastructure needs dedicated central resources but offers an opportunity to simplify the operations and management aspects. Majority of the present Server-based applications can be loosely classified as Servlet-based applications, to be discussed in more details later in this sub-section. Besides this, there are hybrid applications where both client and server functionality are inbuilt in the same app – e.g., SIP-based VoIP apps. Even these applications have moved towards client-server architecture using proxies and gateways to handle the challenges of device mobility, firewalls, dynamic IP, NAT, etc.

For media flow on such platforms, one of the alternatives was to look at media distribution using Content Delivery Networks(CDN). Most of the CDNs today are application specific (i.e., lack standardization) and tend to have security, privacy, and digital rights management implications which lead to high complexity. As CDNs are going deeper, we see a possible integration of SWiFiIC with some of them in future.

To ensure communication in a scenario where the power supply is a significant concern and where end-to-end reachability is not guaranteed, DTN is mandatory. All scenarios, except user and network management, should be able to work even if end-to-end connectivity does not exist. In this respect, it was implicit to rely on a delay/disruption tolerant network (DTN) based platform. Since prior work on DTN and Opp-Net stacks lacked extensibility, we present the overview of Application level "extensibility" of DTN, using IBR DTN as the base for the implementation. We build up from classical client-server paradigm, and discuss mobile computing and recent trends in data centers, before integrating it all for Opp-Nets.

a) *Servlets in Scenario of Internet*

While Internet brought about @@digital connectivity across multiple machines in early 1970s, true impact of Internet started out with World Wide Web (WWW) using HTTP. WWW itself has seen major revisions commonly referred to as web 1.0 (static content from server), Web 2.0 (dynamic content, user-generated content, etc.) and Web 3.0 (personalization and semantics). Client-side dynamic programming using JavaScript in HTML allowed innovations in user experience, but the true transition to Web 2.0 happened primarily on account of server-side scripting. Server-side scripting started with CGI and added a variety of frameworks that allow Java, JSP, PHP or similar language to be used for server-side programming. Concerning Java, such implementation is commonly referred to as Servlet.

Servlet-based programs frequently add another layer of abstraction to have data stored in a database, on a separate machine(s). This leads to three-tier architecture where an end user request from the client machine, is processed by the web-server (commonly called as front-end), using data aggregated from multiple back-end database systems.

Though separation of data from servlet code adds to the software complexity, it allows for tremendous horizontal scaling and faster responses.

Different applications tend to use different physical servers, though this separation boundary is blurring because of co-location of applications (using virtualization) or multi-site implementations on HTTP servers like Apache. Pure servlet based scenario does not work for rural deployments since the remote site may not be reachable to respond.

b) *Mobile computing platforms*

Advent of iPhone in 2007 and subsequent launch of App Store, brought a paradigm change in how applications are developed and deployed. Subsequently, Android and other platforms opened up the smart phone ecosystem and have allowed huge number of affordable devices to be manufactured and sold.

One of the key drivers of the explosion in the applications available on these platforms is tied to the fact that simple applications can be developed for these platforms with minimal effort. Developers of Java and similar platforms were able to move to Android development with some re-training.

To ensure that the applications do not impact user experience, the programming paradigm has changed for mobile platforms to be completely event driven by aspects like activity and service lifecycles, intents/notifications etc. Communication-oriented applications on mobile devices support the ability to work while disconnected from Internet. Applications that support remote sync, e.g., file synchronization or messaging apps can be configured to reduce network usage. Thus they allow for a non-real time communication with servers connected on Internet.

For SWiFiIC applications, the communication targets are other nodes within the local community – hence a need to connect to a server in core of Internet does not arise. The requesting application will be serviced by code running on another device in a non-real-time, asynchronous manner.

c) *Evolution of computers and data centers*

The evolution of computer hardware has followed a shift from centralized computing (during the mainframe) to distributed computing using mini-computers and subsequently to end-user computing using personal computer and mobile devices. The cycle is likely to repeat with the way data centers are developing in enterprises. The initial data centers were located at one site with high-end hardware, and are now seeing a shift towards distributed data centers for fault tolerance and performance (content delivery network, fog or edge computing, etc.). While it appears far-sighted to say that the data centers will move to the premises of end users, given the way content delivery networks are shaping the data centers, they are surely shifting towards the edge of service provider networks and may in future be deployed at village/community level. This is likely to be accelerated by the push for green technologies, to minimize the power in electromagnetic waves and to keep operational costs in control by using less power for the deployments. For example, department of telecommunications, Government of India set a goal of lowering the Electromagnetic field radiation from base station by up to ten times [112]. In future, it is likely that the hub node of SWiFiIC is implemented as a micro or nano data-center collocated at the base station of service provider(s).

Lack of standardization in data centers coupled with specific needs of different applications has implied that the data centers need skilled professionals to operate them. To deploy data-centers at community level, there is a tremendous need for standardization and simplification. Initiatives like “Open Compute” have tried to standardize the data centers for large deployments. They may expand their scope to standardize the edge computing infrastructure for 5G networks and beyond. Some of those platform components can be used by SWiFiIC in future. The residual complexity will need to be abstracted from the SWiFiIC operator and end users.

For this sub-section, most of the communication will stay limited to local community. With data centers and cloud-like infrastructures being used for backup / archival etc. and possible future extensions of SWiFiIC hub.

d) *DTN Routing and Applications*

A DTN session layer has been proposed by Demmer et al. in [113]. In related prior work [114] simulations have been done using NS2 by Jain and others, targeting implementation of delay tolerant network.

As discussed in 2, DTN routing itself can be accomplished by participating mobile nodes (commonly referred as Independent Mobile Nodes based routing) or using auxiliary nodes for routing. Early routing efforts relied mostly on epidemic flooding [86]. Later work started using utility and priority based flooding [8] or using past encounters to find optimal nodes (e.g., PRoPHET[115]). More recently multiple routing approaches have been proposed using behaviors of the entities that carry the device participating in DTN. For example in context of rural communication almost all the devices will be carried by people. In this perspective, social properties of the person carrying the device comes into routing decisions. SimBet[116], BubbleRap[117] and M-Dimension[118] are some of the social routing approaches studied by the authors.

The social routing aspects are predominantly evaluated using simulations and are a challenge to integrate into real-world scenario. Since each of the nodes cannot have a global view of other nodes, social routing approaches do not scale well for large deployments. Moreover, some of the routing approaches include location and community information as part of node identification, causing a challenge for tracking node identity. In future, SWiFiIC targets to compensate for this by tracking the node characteristics similar to the study in 5.

For SWiFiIC we have not proposed approaches for routing optimizations, and we expect it to be handled on a need basis. If the operator of SWiFiIC sees an opportunity to provide better service, they may decide to deploy auxiliary nodes like throw boxes and mules connecting them to the hub. Alternatively, the operator may extend the reachability of hub’s wireless access using wired infrastructure. IMN involving regular end-user devices with epidemic flooding is used in the current deployment.

We have used IBR DTN as the protocol stack implementation for SWiFiIC. Based on experiments in developing applications on IBR DTN for communication needs in rural environment, three drawbacks were noticed. First, it is not easy to adopt a new application; second, it does not have features like management and control; and third, it does not have a simple user interface to connect Android devices using WiFi-Direct.

It is in the perspective of the first two problems that we decided to develop an eco-system of hub with multiple nodes talking to the hub. As observed in 5, this approach increases the communication cost and delay, but it allows for quick application development and conformance to legal needs like lawful interception, etc. The present design is implemented around a single hub. Future changes will allow multiple hubs in the eco-system.

e) *Deployment architecture options: Pure distributed (multi-hub) vs.Centralized vs. Nodes and Hub*

Three alternatives were considered for SWiFiIC platform a) pure-distributed vs. b) central server vs. c) hybrid architecture. Many peer-to-peer networks are capable of operating in a pure distributed manner – and similar logic could be evolved for SWiFiIC. IBR-DTN [44] stack (from Braunschweig University of Technology) in its current form is implementing this approach. Similarly, the client-server model can be implemented in a manner that all end devices are clients and central entity serves the requests. This limits the application scenarios to hardware capabilities of the central entity, and hence scalability will be affected. The third alternative is a hybrid scenario where central piece acts merely as a relay connecting the server and clients – while the server and client components are hosted on mobile devices. To visualize the third alternative in a different aspect, it can be crudely compared with Ad-Hoc WiFi vs. Access Point based WiFi.

We chose a hybrid approach involving Nodes and Hubs for following reasons:

- Pure distributed systems tend to lack ownership and are extremely difficult to manage and monitor.
- Sustainability needs revenue generation: the operator of the hub has an opportunity to monetize the overall deployment. For pure distributed systems, monetization will involve more complicated implementation.

- The current design does not preclude multiple hubs to be defined. If the real world deployments suggest that the monetization can be done with multiple hubs, the solution can be architected for the same.
- Hub allows for quality to be monitored and coverage gaps to be identified.
- And extended implementation of Hub can drive routing decisions.
- For integrating with worldwide Internet, hubs can host the gateway pieces.
- Because of hybrid nature, hub can act as server for some of the applications.

7.2.2. *SWiFiIC Design*

For simplicity, the discussions below assume a limited deployment of SWiFiIC. Some of the complicated design elements have been intentionally skipped. SWiFiIC platform targets four primary goals

- Simplicity of deployment and easy management.
- Usage of “off the shelf components” to benefit from commoditization to achieve cost efficiency and minimize operational cost.
- Commercial model to drive revenue generation.
- Extensible platform with ability to add applications.

To achieve these goals, a deployment of SWiFiIC uses a Linux based hub to control the deployment. All end user devices are Wi-Fi enabled smartphones or tablets. All the communication uses Wi-Fi (either as Wi-Fi direct or assisted by consumer-grade access points).

The platform allows specific applications to have gateway/proxy capabilities extended using the Hub. E.g., broadband connectivity (like NOFN) to the village can be integrated into the Hub. In the absence of wired or optical connectivity, 3G/4G can also be used for connectivity. If the cost of communication is a concern, these gateway elements can compress the information. Moreover, the platform allows for integration of DTN as transport for open source frameworks like IoTivity[119]. As of now, the hub is not being used as a gateway for providing Internet-based services. It only show-cases the communication within the village.

a) *SWiFiIC Physical entities:*

App Hub: Rugged Portable Linux System with Tomcat, IBR-DTN implementation, MySQL, munin and multiple SWiFiIC Hublets for the deployed applications. The hardware will generally be power backed up with small battery; solar powered if feasible and include WiFi (802.11n or better). It may be periodically connected to Internet using Fiber / ADSL / 3G / 4G modems. Present version runs Ubuntu 16.04.4. Linux has been chosen primarily to aid debugging during the initial prototyping phase. If needed, it can be moved to an Android device in future. The cost of the device is INR 12,000 (in 2013).

UT: End users have access to SWiFiIC platform using Android devices. All communication relies on Wi-Fi and happens using IBR-DTN service running on these devices. SWiFiIC user terminal is a portable Android device running Android 6.0 or higher. It runs SWiFiIC Apps deployed from App Hub using a custom SWiFiIC UT Application (SUTA). Extensible applications make use of data provided by SUTA to further implement their functionality. E.g., Messenger gets the list of users.

Operator Terminal: With the intent that App-Hub need not be physically exposed, an Android device connects to services running on the Hub to provide all management and troubleshooting activities. E.g., User addition and modification, account recharge, etc. are provided by the SWiFiIC Operator App (SOApp) running on the Operator Terminal. SOApp does not rely on DTN. Instead it directly communicates with the Hub over HTTP.

b) *SWiFiIC Logical entities:*

There are platform specific components called SUTA and SOApp. Other than these, SWiFiIC allows third-party applications to be developed and deployed on the platform. Each independent App has two components – 1) Android application and 2) Hub specific code. In SWiFiIC paradigm, the third party app is called Swilet. The hub component of Swilet is called Hublet, while the corresponding mobile applications are referred to as Moblets.

SUTA Moblet running on end-user device, provides access to the Swilet. As a new Swilet is deployed, the Hublet component automatically runs on the Hub. The details of the new Swilet are pushed (multicast) to all SUTA instances. Once the end user is notified, the end user may decide to install the Moblet piece through SUTA. This functionality is still under development.

Periodically, the SUTA Hublet multicasts the list of users to all SUTA Moblet instances. The SUTA Moblet also responds periodically to these heartbeat messages to provide feedback on round-trip delay, node reachability, etc. The response includes statistics of device like battery power, software version, etc. This information is used by Hublet to update the list of active users and their device details.

The Moblet for other apps can query SUTA for SWiFiIC deployment-specific information. Presently this is limited to details of Hub and the list of end-users in the SWiFiIC platform.

Swilets: This is the application entity that will be deployed on SWiFiIC. It is a combination of SWiFiIC aware Android application and servlet with DTN adaptation. The android application is called End User Application (EUA) and the DTN adapted code running on hub is called hublet. Besides these two entities, Swilet also contains a swilet-manifest.

SUTA[SWiFiIC User Terminal App]: Any SWiFiIC user terminal needs SUTA to be installed on the device. SUTA allows access to SWiFiIC marketplace hosted on the App Hub. SUTA user interface allows end user to do common self-service tasks like balance query, user lookup, etc. It also wraps the IBR DTN service to hide DTN specific complexities from the end user. For management, monitoring and research specific data collection, SUTA periodically pushes device metrics to AppHub. It also acts as content provider to other EUA for user lookup etc.

DPDU: This defines the protocol data unit used by all EUA and Hublets. They are either Action request from EUA to Hublet or notifications to EUA from Hublet. For multimedia scenarios, the DPDU includes file content. The payload is XML based and can be compressed for saving bandwidth.

Hublet: This is the DTN aware servlet code that acts on DPDUs. In majority of the usecases, this is a thin layer of code that converts Actions to Notification and relays to different user terminals based on the application logic. It also includes updates to databases on the Hub for some of the applications.

Following additional components are part of SWiFiIC software that runs on the hub:

- SWiFiIC Portal – allows synchronous access in browser to download SUTA APK and serve other static content.
- Operator Servlet – backend for SOApp, provides manageability interface for SOApp app discussed subsequently.
- SWiFiIC database and Hublet database(s).
- SUTA Hublet: this provides the DTN based App Hub functionality for SUTA. The data collected by SUTA Hublet is analyzed by Operator App (SOApp).
- App Hub Utilities: These are set of code that periodically audits the system, does the billing, generates management reports, etc. In future, it will expand to cover App Hub upgrade and App hub migration to a new Linux system, backup, etc.
- IBR DTN: The DTN daemon running on the hub.

*c) SWiFiIC Operator App (**SOApp**):*

This is the management user interface application for SWiFiIC running on a portable device. Current functionality includes provisioning a new user, user account recharge, billing, etc. The future management scope includes app-hub resources, deployed hublets, users, and user terminals. / UT, manage existing hublets on app-hub, resource monitoring on app-hub, tracking DTN routing efficiency, UT metrics analysis, etc.

d) SWiFiIC communication

Except for the SOApp, all communication in SWiFiIC relies on unicast or multicast based Delay Tolerant Network. In the present iteration, all Moblets talk to their respective Hublets (unicast). The Hublet code may relay or reply in unicast or multicast manner.

As mobile devices come in contact with each other, they exchange the messages so that they reach the destination. Because of reliance on DTN, many DTN bundles may not reach the destination or have significant delay (order of hours). The Swilets need to implement their own logic (like sequencing, acknowledgment, retry, etc.) to ensure that it gets the level of quality they desire. Occasionally the end user may need to be notified, especially for people who live on the periphery of the village.

e) *Cost and benefit of Centralized Hub*

All messages are relayed to the Hub and then distributed back towards the recipients. This approach was necessary to ensure monitoring (and hence billing) of the setup. In future, this will also help scan the messages to include IDS/IPS functionality. Further computationally heavy tasks or large storage of data can be delegated to Hublet, thus reducing the space and power needs on mobile devices. Hublets can also act as gateway/proxy to Internet.

Because of such a centralized communication, round trip delays will increase, and delivery ratios may drop. In separate ongoing work, authors have done simulations to measure such impact. It was found that in the vicinity of the Hub, congestion sets in as load is increased. To mitigate this, in future design, the team intends to have multiple hubs directly connected over wire.

One benefit that the centralized design allows is the ability to measure quality of service for end users. If the SWiFiIC operator finds that some devices are having poor connectivity, she/he may decide to use dedicated ferry based mechanism to serve such nodes. For example, they may hire a person to periodically visit the area where the poorly served nodes reside, or they may allow higher replicas to be created for messages originating or terminating at such nodes.

f) *Base Platform Components*

SOApp and SUTA are two crucial parts of the SWiFiIC platform. Moreover, libraries are provided for Hublet and Moblet pieces to simplify development. Figure 7.1 below captures the deployment for Hub. Figure 7.2 captures the deployment for the typical end-user terminal.

g) *Hublet basic logic*

Hublet code relies on functionality provided by generic code within the Hublet Adaptor to mask the complexity of communication. The hublet implements the mapping functionality on App-Hub for the swilet. Based on the content of incoming DPDU, an outgoing DPDU may be generated and/or Swilet specific database may be modified. Helper libraries are provided for updating the hublet database, file storage etc. For each Swilet, the Hublet needs to implement the handlePayload function. This is the code that gets triggered on receipt of message from Moblets.

Algorithm 7.1: Handle Payload

```
function handlePayload(payload, context)
    Identify the operation within the payload;
    Check account privileges of the originator of payload;
    Do the necessary billing for the operation;
    Invoke the implementation of operation;
End
```

handlePayload is the single entry to process all types of operations for the specific Swilet. E.g. for Exam Swilet, it handles operations like, addStudent, sendPaper, submitAnswerScript, getAnswerScripts, getAttendance etc.

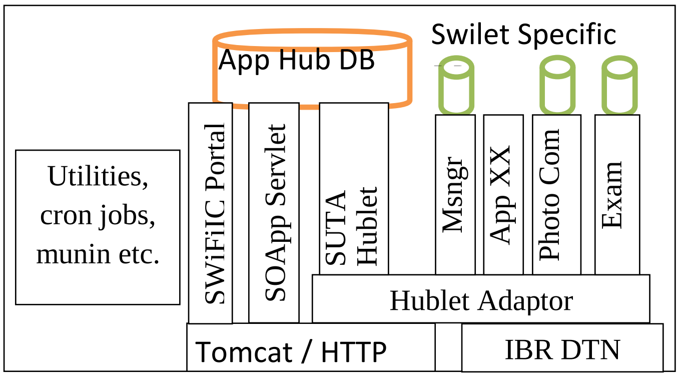


Figure 7.1 Deployment on Hub

A check on account privilege is necessary to verify the user and avoid impersonation. Billing aspects are optional. The implementation of operation is Swilet specific logic and generally operations such as queries to Swilet specific DBs and generation of DTN messages back towards Moblets.

Future extensions can allow Hublets schedule cron jobs etc. by declaring the same in Swilet-manifest. App-Hub components will also be able to monitor hublet's resource utilization metrics like db-schema size, file system usage, DTN load generated, usage of Internet link and CPU load.

h) *Moblet basic logic (Android apps)*

The Moblet's in present context are Android applications. Other than implementing the logic for the user interaction and application specific requirements, the Moblet's implement a broadcast receiver to handle incoming messages from the Hublet. Further, they rely on base libraries (SWiFiIC JAR) to register with the IBR DTN service and to send/receive their messages. End-user application uses Android's SDK along with SWiFiIC java library for DTN Communication. This library also includes helper routines to create/parse DPDU.

The main activity in the EUA invokes SWiFiIC. Initialize ("App Name"), and the Android manifest defines intent-filter for all activities that will handle notifications delivered by SWiFiIC. Such activities generally extend from DTNActivity which in turn extends Activity as defined by Android SDK.

For getting the context of SWiFiIC deployment like Hub's DTN address, list of users, etc., Moblets use content provider interface of SUTA.

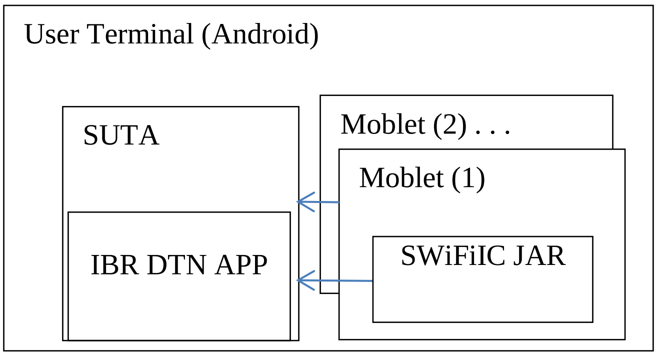


Figure 7.2 Deployments on User Devices

i) *SUTA moblet design*

SUTA serves three functionalities: i) UI - the user's interface to SWiFiIC; ii) Interface to explore and downloads of APKs (moblets)f; iii) Content provider to other EUA for generic SWiFiIC functionality – e.g., list of users, balance, etc. The user interface to SWiFiIC provides SWiFiIC marketplace access, billing details and performance details like reachability of hub, etc.

Content provider functionality of SUTA allows keeping the UT user list in sync with app hub. In future, it will allow query of Swilets details, get role/access details for UT concerning specific swilet, query statistics on expected quality of service etc.

7.2.3. *SWiFiIC Applications in action*

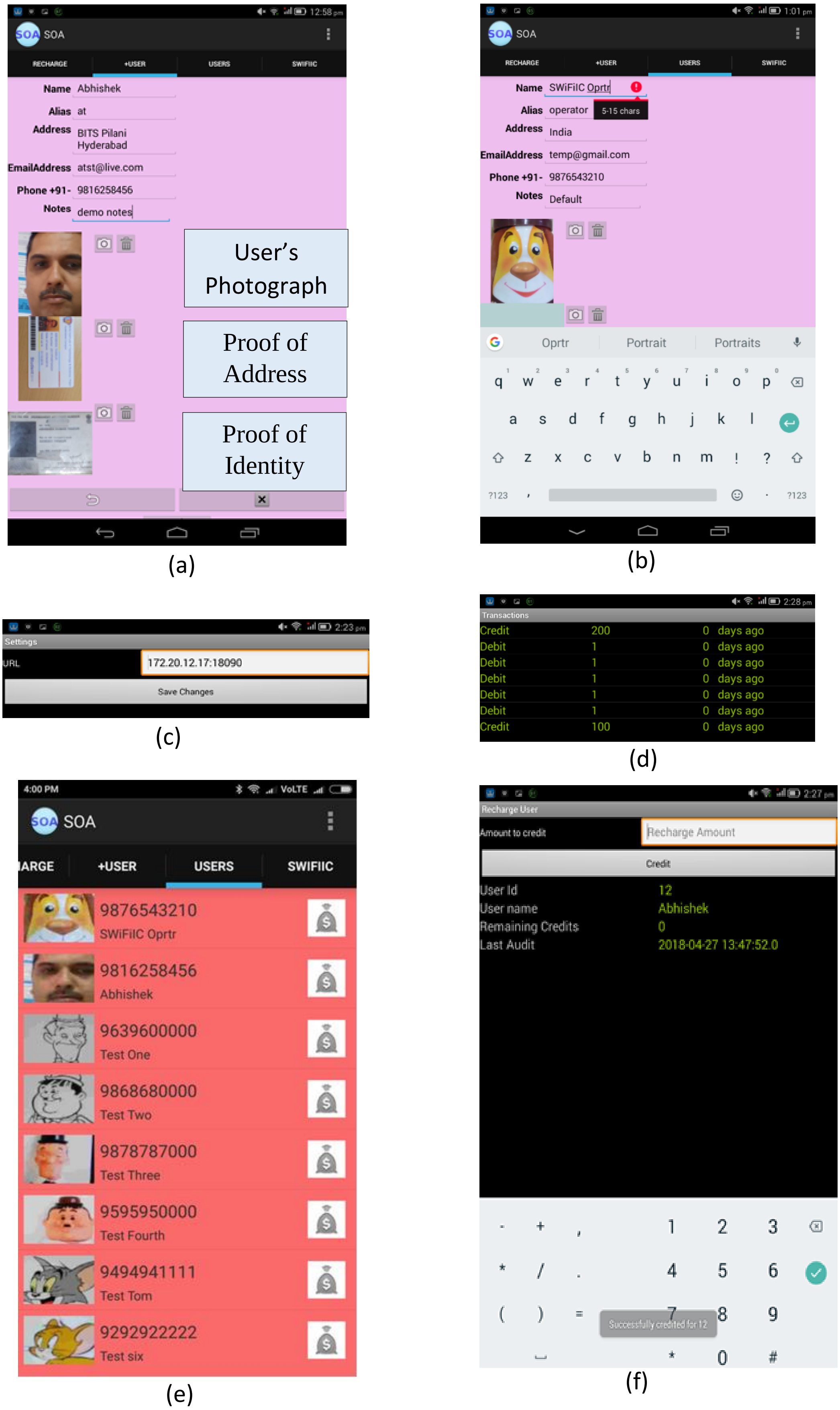
In this section we study the design and deployment details for some of the common applications. Deployment screenshots are included to showcase the user experience.

a) Operator Application: User Addition

SWiFiIC Operator App (SOApp) is an Android application that relies on HTTP request/responses to App Hub (Operator Servlet interacting with SUTA Hublet and App hub utilities).

This app uses synchronous network communication and enables management of the SWiFiIC ecosystem. "User Addition" activity within this app minimizes the user management effort. It captures all details of a new customer and populates the same in the AppHub schema.

The SOApp app captures the user details (including pictures for identity proof, device MAC address etc.) and uploads on AppHub. Once the upload is successful, a message is multi-casted from Hub to all SUTA instances. Figure 7.3(a) captures the operator adding a new user (Abhishek) while Figure 7.3(b) is for updating a user (SWiFiIC Operator, to a new image and name as SWiFiIC Oprtr). Subsequent figures show settings for the hub, details of user billing, user list and recharge interfaces. Fub-figures (c-f) cover settings and additional screenshots.



12:58 pm

SOA

Name Abhishek

Alias at

Address BITS Pilani
Hyderabad

EmailAddress atst@live.com

Phone +91- 9816258456

Notes demo notes

(a)

1:01 pm

SOA

Name SWiFIIC Oprtr

Alias operator 5-15 chars

Address India

EmailAddress temp@gmail.com

Phone +91- 9876543210

Notes Default

(b)

2:23 pm

Settings

URL 172.20.12.17:18090

(c)

2:28 pm

Transactions

Credit	200	0 days ago
Debit	1	0 days ago
Debit	1	0 days ago
Debit	1	0 days ago
Debit	1	0 days ago
Debit	1	0 days ago
Credit	100	0 days ago

(d)

4:00 PM

SOA

9876543210
SWiFIIC Oprtr

9816258456
Abhishek

9639600000
Test One

9868680000
Test Two

9878787000
Test Three

9595950000
Test Fourth

9494941111
Test Tom

9292922222
Test six

(e)

2:27 pm

Recharge User

Amount to credit Recharge Amount

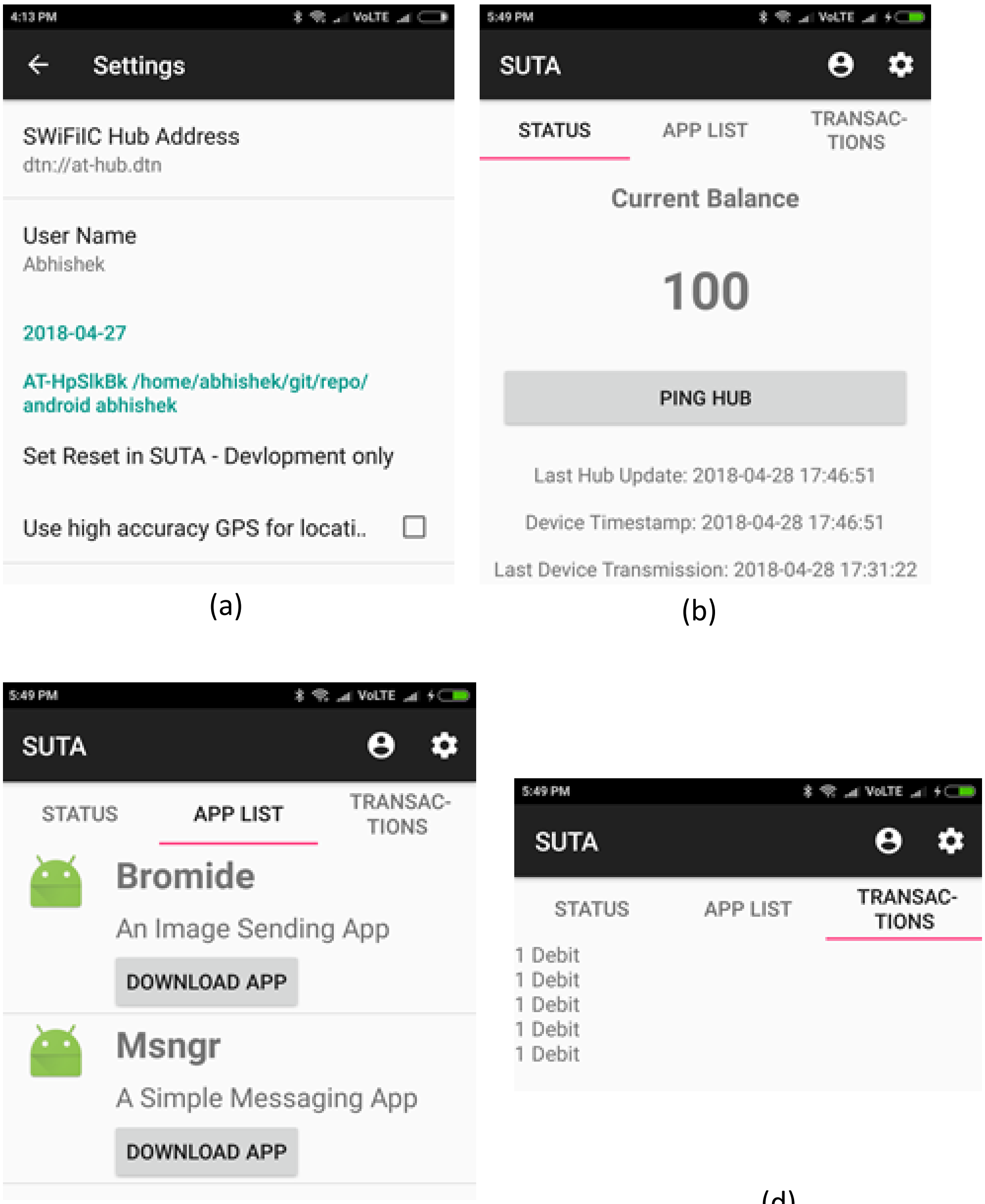
User Id	12
User name	Abhishek
Remaining Credits	0
Last Audit	2018-04-27 13:47:52.0

Successfully credited for 12

(f)

Figure 7.3 SOApp user addition and user update screenshots

b) SUTA: user terminal App



4:13 PM

← Settings

SWiFiIC Hub Address
dtn://at-hub.dtn

User Name
Abhishek

2018-04-27

AT-HpSlkBk /home/abhishek/git/repo/android abhishek

Set Reset in SUTA - Devlopment only

Use high accuracy GPS for locati.. □

(a)

5:49 PM

SUTA

STATUS | APP LIST | TRANSAC-TIONS

Current Balance

100

PING HUB

Last Hub Update: 2018-04-28 17:46:51

Device Timestamp: 2018-04-28 17:46:51

Last Device Transmission: 2018-04-28 17:31:22

(b)

5:49 PM

SUTA

STATUS | APP LIST | TRANSAC-TIONS

Bromide
An Image Sending App
DOWNLOAD APP

Msngr
A Simple Messaging App
DOWNLOAD APP

(c)

5:49 PM

SUTA

STATUS | APP LIST | TRANSAC-TIONS

1 Debit
1 Debit
1 Debit
1 Debit
1 Debit

(d)

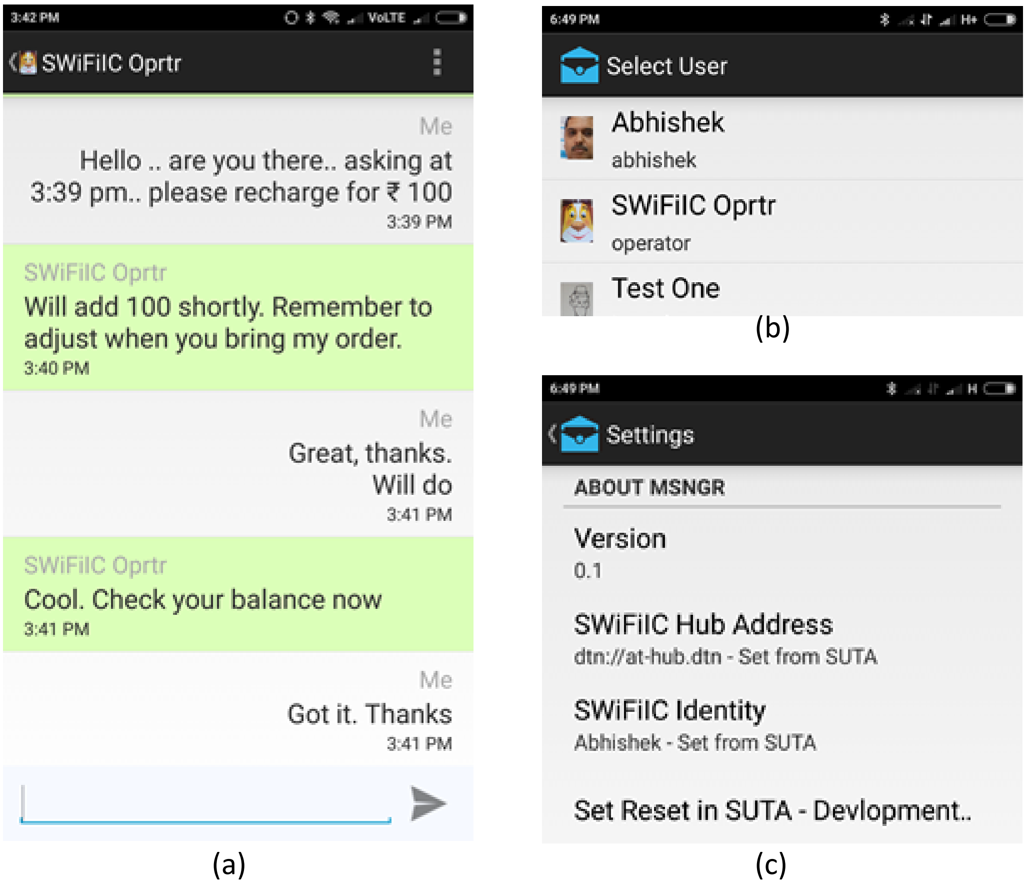
Figure 7.4 SUTA Moblet

Figure 7.4 captures screenshot of SUTA application in action. Figure 7.4(a) shows the settings where the address of hub and user identity can be updated. Main tab of the application shows the user account balance communication timestamps for the SUTA moblet and hublet. Second tab shows the list of Android apps. They can be downloaded using DTN. Third tab shows the last five transactions for the user account.

Hublet: The SUTA Hublet code provides user list, applications available on hub, basic debit details to user account, present balance, etc. It generates periodic updates to a multicast endpoint so that new devices also get this information. This multicast also includes the account details (e.g., balance for each account, last five debit transactions, etc.) for each user. Image of the newly added users is also shared by the Hublet.

Moblet: The mobile application periodically sends a heart-beat DTN message to the hub. This helps to get an estimate of round-trip-time between the hublet and moblet.

c) *Messenger*



3:42 PM

SWiFiIC Oprtr

Me
Hello .. are you there.. asking at 3:39 pm.. please recharge for ₹ 100
3:39 PM

SWiFiIC Oprtr
Will add 100 shortly. Remember to adjust when you bring my order.
3:40 PM

Me
Great, thanks. Will do
3:41 PM

SWiFiIC Oprtr
Cool. Check your balance now
3:41 PM

Me
Got it. Thanks
3:41 PM

(a)

6:49 PM

Select User

Abhishek
abhishek

SWiFiIC Oprtr
operator

Test One

(b)

6:49 PM

Settings

ABOUT MSNGR

Version
0.1

SWiFiIC Hub Address
dtn://at-hub.dtn - Set from SUTA

SWiFiIC Identity
Abhishek - Set from SUTA

Set Reset in SUTA - Devlopment..

(c)

Figure 7.5 Messenger App – Multiple screens

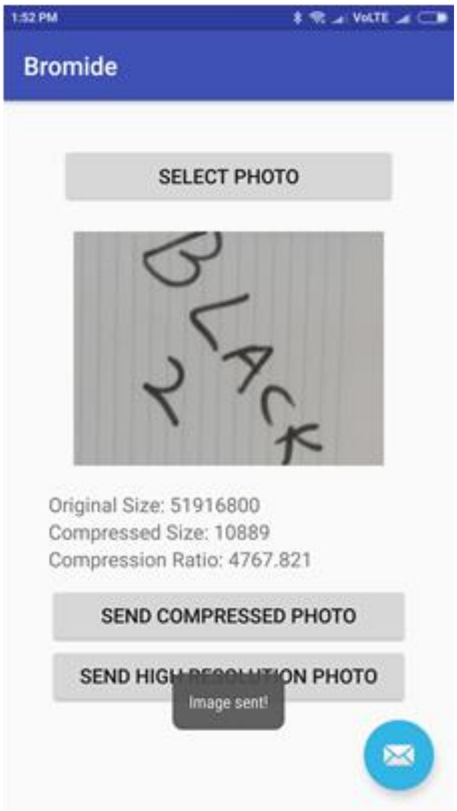
Figure 7.5 shows a sample chat session, user selection list while creating a new message and settings for the Messenger app. The captures are done for user “Abhishek” logged into the SUTA app.

Hublet: The messenger Hublet after authenticating the sender, debits (notional amount of 1 rupee presently) from their account. After that it queries the DTN id for the receiver of the message and relays the message to the destination DTN node.

Moblet: Majority of the code here is related to UI and tracking of messages received in the past. Also, using the Android's broadcast receiver interface, the code handles incoming DTN messages. To send a new message (towards Hublet) the code uses sendAction on Helper functionality provided by the SWiFiIC JAR files.

d) *Image upload app*

Bromide, a sample app to demonstrate capability to upload images was developed. The moblet uses the SUTA interface to query the address of hub and thereafter sends the image file (either compressed or raw format) to the hublet based on user choice. Hublet simply stores the file in a preconfigured location.



Original Size: 51916800
Compressed Size: 10889
Compression Ratio: 4767.821

Image sent!

Figure 7.6 Image upload App

7.2.4. *SWiFiIC - Observations and Future Work*

SWiFiIC design is a novel approach to deploy applications for rural usage. The underlying concepts themselves have been borrowed from existing work, but the extensibility of the platform and its ability to operate in delay tolerant environment is something that was not attempted before. The simplicity of application development, management and low cost of operation are other differentiating factors of SWiFiIC.

The message delivery has been limited and worked well only in the presence of specific access-points. For the period of pilot run (April 2014 for three days), almost half the messages were getting dropped when epidemic routing with one-hour bundle lifetime was used. While some devices had support for WiFi-Direct, the implementations in Android was observed to be lacking in usability. During the trials for SWiFiIC, it has been noticed that user experience for Wi-Fi Direct is suboptimal. It confuses some of the end users (by popping up frequent messages). As the WiFi-Direct implementations improve in next releases of Android, we expect the delivery rates to improve as well.

The peer discovery for IBR-DTN relies on multicast support. In many scenarios for Android devices, the peers have not been discovered even after few minutes. Restarting the Wi-Fi interface or restarting the IBR-DTN service frequently solves this issue. It was also observed that some access points allow peer discovery to happen most of the time, while a few do not.

Application development feedback was collected using undergraduate students as app developers and end users in the academic institute. It showed significant training and development time was needed for developing even the basic applications (three weeks against the goal of five days). For the set of students that had developed Android applications in the past, the effort was significantly less (around ten days for messaging apps). Based on feedback from the students, this was attributed to their exposure to “procedural” and “object-oriented” coding which did not involve asynchronous tasks. The framework as trialed needs more tools to support debugging and integration.

Based on present observation, rather than sending all communication via the hub, system has to be redesigned to send unicast messages directly to the destination, while mandating multicast and broadcast messages to go via the hub. All logs related to creation and delivery of bundles will still be sent to the hub. User Interface design is another aspect where the resultant apps can be improved. The student team was not able to appreciate the usability aspects.

Implementation of real-world routing decisions using knowledge shared by the hub another area to be explored. Present implementations have shown that social aspects of DTN routing are too large to adapt to real-world devices. The authors target to reduce complexity by allowing the Hub to handle most of the complexities and limiting the devices to explore routing decision for just a handful of devices.

The authors aim to explore the options of connecting multiple app-hubs with the possibility of a single UT belonging to multiple app-hubs. Such a scenario will bring in additional challenges for the framework. Communication privacy and security aspects are yet to be experimented with. Field trials are expected at the end of the year, which will truly determine the success of the framework or expose possible shortcomings. True rural deployments will also provide learnings that need to be incorporated back into the platform.

SWiFiIC platform is being operated by multiple students within the campus of the authors. It is still far from production ready. Some of the pending items are

- Providing a better way to use Wi-Fi Direct. Android Nearby is a promising approach. But it is not integrated into DTN stack yet.
- Developing other apps based on needs identified by rural people.
- Deployment in true rural scenario.
- Multiple Hub with wired connectivity to avoid single point of failure and reduce congestion.
- Auditing mechanisms on the hub so that a Hublet does not deny services to other apps.

In future we may implement further functionality. The project code is maintained as open source at www.swifiic.in.

7.3. System II - Vectors

SWiFiIC with limitation of WiFi-Direct and other discovery failures was found to be inadequate for long term streaming of user-generated video content. The availability of the Nearby API[45] in Android, provided us an opportunity to deploy a system for streaming video in such scenarios. To ensure that the network resources are adequately used, we combined Scalable Video Coding (SVC)[47] and adapted the communication such that video flow does not overload the network and at the same time attempts to improve the quality at the receiver if possible. Thus we created a simpler version of what is proposed and simulated in 3.

Our key contributions as part of this demonstration involve – splitting and joining of SHM encoded content; SNW like relay from source using Nearby based communication to create a DTN-like network, and feedback and adaptation to optimally use the network resources.

The Nearby API [45] works by using a combination of Bluetooth, Wi-Fi Hotspot, Wi-Fi Direct, and location data to establish the connection between two devices. This API accomplishes the creation of P2P connections between nodes without user interaction. As IBR-DTN does not support connecting over Android Nearby, we have implemented a lightweight DTN-like protocol named “Vectors”. It uses the concept of binary Spray-and-Wait [14]. Instead of implementing the Bundle protocol [48], we have used JSON files for storing and tracking the payload metadata. Vectors use Bluetooth for communication. As observed in 6, Bluetooth provides considerably better energy efficiency than Wi-Fi.

However, the choice of communication protocol alone is not enough for optimal adaptation of video flow. We also need a way of prioritizing the video payloads so that at least a basic quality is achieved at the destination.

Similar to 3 Vectors: 1) carries scalable video chunks over opportunistic contacts; 2) uses of different replication counts depending on the scalable video layer; 3) generates cumulative acknowledgment for better feedback and adaptation. In deviation from what is covered in 3, Vectors in its present form only adapts the operating point. I.e., it does not control TTL and does not reduce the replication count for higher layers. The routing does prefer higher copy count and deletes delivered bundles, but is not media aware. To summarize using acronyms of the ORT and HDM parts on R and HD are implemented in present deployment of Vectors.

7.3.1. *Vectors Design*

Each SVC layer is extracted into separate payloads by the source. The source also transmits this extraction information to the destination using Vectors. The decoder combines the received payloads using the extraction information. Note that the lower layers and the extraction information are necessary to decode higher SVC layers.

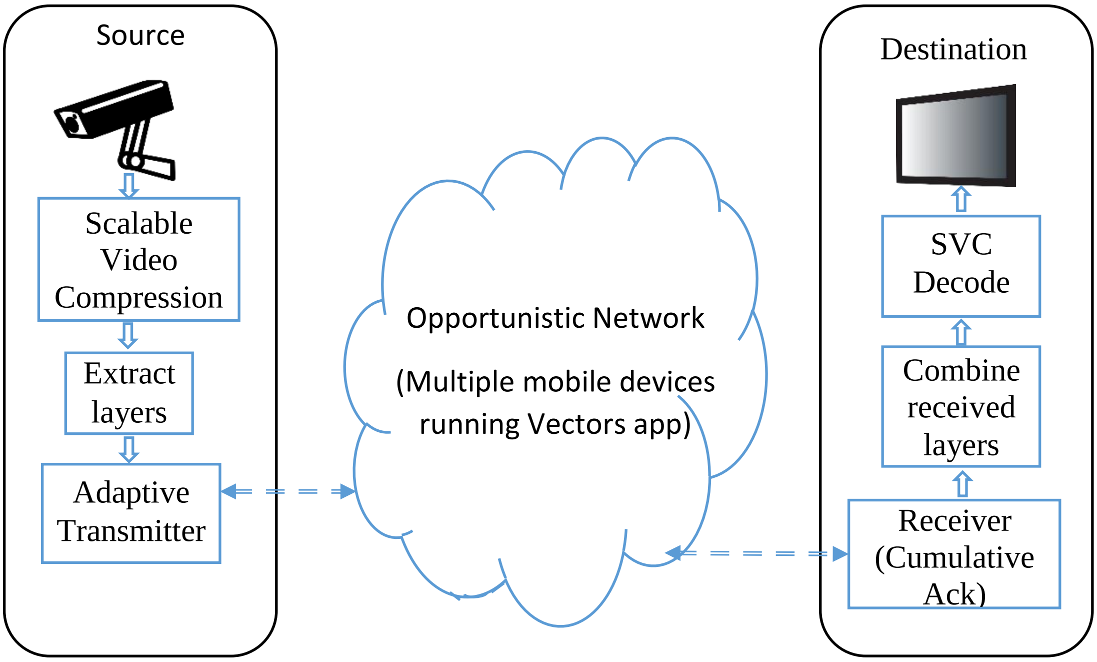


Figure 7.7 Data flow for SVC content using Vectors

Figure 7.7 shows the high-level design of the Vectors streaming platform. Content from capture deice is compressed using scalable video coding. We then extract the layers and transmit them to the Nearby based Opp-Net using an adaptive transmission algorithm. At the destination, end receiver generates acknowledgments for the received payload and also passed the chunks to be combined before decoding. For our deployment, both source and destination are Linux based computers since scalable compression API was implemented on Linux platform. The communication with Opp-Net relies on android-debug-bridge (ADB) to push the content to mobile phones over USB interface. In future, if the modules are available on Android platform, mobile phones can be used as source and destination.

Subsequent Figure 7.8 captures the design details of the Android application that carries the video chunks

As the connections between mobile devices running Vectors are opportunistic, the connection could be terminated any time due to the two devices moving out of range. Hence, the file transfers can be considered to be best effort. Our goal is to maximize the number of successful SVC layer transfers between two connected nodes given the constraints of available bandwidth and connection duration between devices participating in the opportunistic network.

a) *Communication state in Vectors*

The communication activities including advertisement, discovery, connection management, and files transfers are implemented in an Android background service. The state of communication is captured in Figure 7.8.

Discovery: Nodes broadcast their ID using Bluetooth Low Energy based communication [https://source.android.com/devices/bluetooth/]. Once a node discovers another node, it checks the ID of the node for whether a connection can be established. The node then initiates a connection using the Nearby protocol. If the connection can be successfully established, the node moves to Connection Initialized state. Otherwise it remains in Discovery, attempting to connect to other nodes.

In case a discovered node has been recently connected it (last few minutes), the connection is rejected. This is done to improve the chance of discovery of newer nodes. This also avoids spurious connection since both the nodes are likely to have the same set of SVC layers if they had recently communicated.

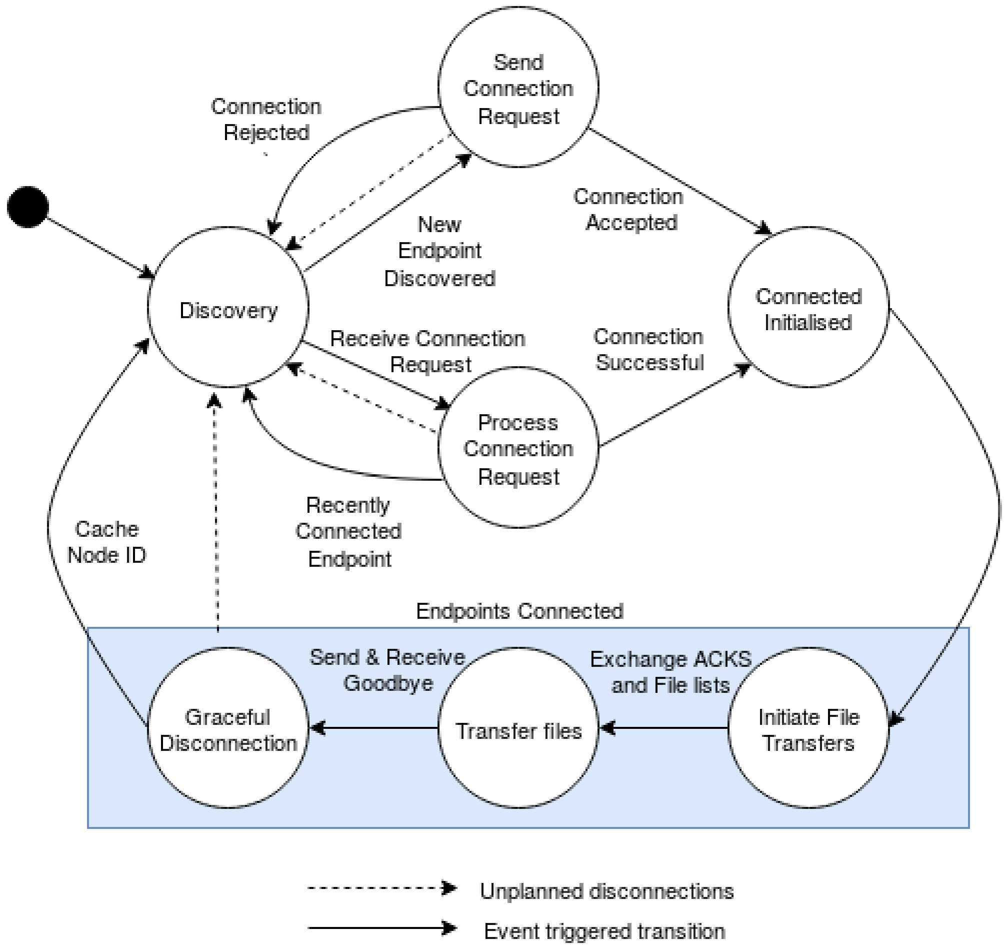


Figure 7.8 Vectors Connection States

Connection Initialized: Once both the nodes have connected to each other, messages are exchanged to setup the transferring of SVC layers. Each message relayed contains a four-byte header with the type of message followed by the message payload.

Endpoints Connected: To begin with, the Acknowledgement (ACK) message from the video destination is exchanged between the devices. The ACK message contains the timestamp of when it was created by the destination. If the received ACK is newer than the current one on the device, the ACK is replaced with the newer version. The ACK file works on a selective acknowledgment basis with the list of the SVC layers and metadata files received by the destination. Files present on the device which have been ACK’ed by the destination are deleted from the device.

In the next step, the nodes exchange the list of the files they have. Each device then computes the list of files it needs and requests for the same. The devices then exchange the files after checking for the Time To Live (TTL) expiry and ordered by replica count(L) held by the nodes.

Each file is successively transferred with its Vectors metadata file. The metadata file tracks the value L of the file and the nodes the file traversed. The value L is halved on each hop between devices to conform to the binary Spray-And-Wait algorithm.

Connection Termination: Once the two nodes have successfully transferred all the files with each other, they will send short goodbye messages to inform the other that it has received all the files successfully. Each node stores the other node's ID and the time of disconnection in a list. As discussed during discovery, a node will not attempt to reconnect to any other node in this list for one minute.

b) *Design of Source: Usage of encoder and splitting of layers*

For our experiments, we use SHVC (Scalable High-Efficiency video coding) for video compression. The scalable approach is used so that we can adapt to the network bandwidth based on the feedback received from the destination. SHM [120] is used to with GOP size of 16. This provides for five temporal scaling layers. We also add a special enhancement layer.

The individual video coded layer (VCL) from SHM (12.4) encoded content is extracted using custom code developed by us. As part of extraction, we also create the metadata for combining these VCL at the destination. Both the base-layer VCL and metadata for the video chunk are transmitted using a large value of L.

For the campus deployment, we have kept a delay target of twenty-four hours for playout. Further video chunks are captured every 12 minutes. As part of adaptation, we check if the content for six, twelve and twenty-four hours earlier rounds have been acknowledged. If none of the chunks are acknowledged, the adaptor does a multiplicative decrease for next chunk (10% drop). If one or more chunks of these three chunks are acknowledged, additive increase is done by 1%, 2% or 4% depending on status for the three rounds.

c) *Design of Destination: Ack, Combining the layers and decoding*

On delivery of each payload, a new cumulative acknowledgment with last thousand received payloads is generated. Periodically the delivered files are combined using the extraction metadata. If either the base layer (L0T1 in our experiments) or extraction metadata is missing the video chunk cannot be decoded at all. If any of the higher enhancement layers are received, but lower enhancement layers are missing, the higher layers cannot be decoded.

Missing frames for temporal layers are added by repeating the prior frame. To compensate for spatial and quality scaling (L1TX), the lower resolution frames are scaled up using DownConvertStatic tools from JSVM (9_19_15). For ease of playback and to conserve storage space, the decoded YUV frames are re-encoded using ffmpeg.

7.3.2. *Experimental setup and results:*

Since SHM based compression/decompression needs computation resource significantly and in the absence of other SHVC implementations on mobile platforms, we used Linux based nodes for video capture, encoding and decoding. As mentioned above, the opportunistic network is created using Nearby based implementation running on multiple devices with Android Marshmallow and higher. The code base is made publically available at Github [note as of May 2nd 2018, the codebase has not been exposed]. Figure 7.9 captures the geographical layout of the setup.

During experiments, 15-20 nodes had vectors installed on them. This is the typical density that we can expect in a rural scenario. The library is visited by few of the nodes during the day. As the students return to hostel (some of them stayed in Gandhi Bhawan), they complete the delivery of the payload. The acknowledgments tent to get delivered the next day. Note that for period of experiments, library had extended visiting hours hence we saw slightly better performance.

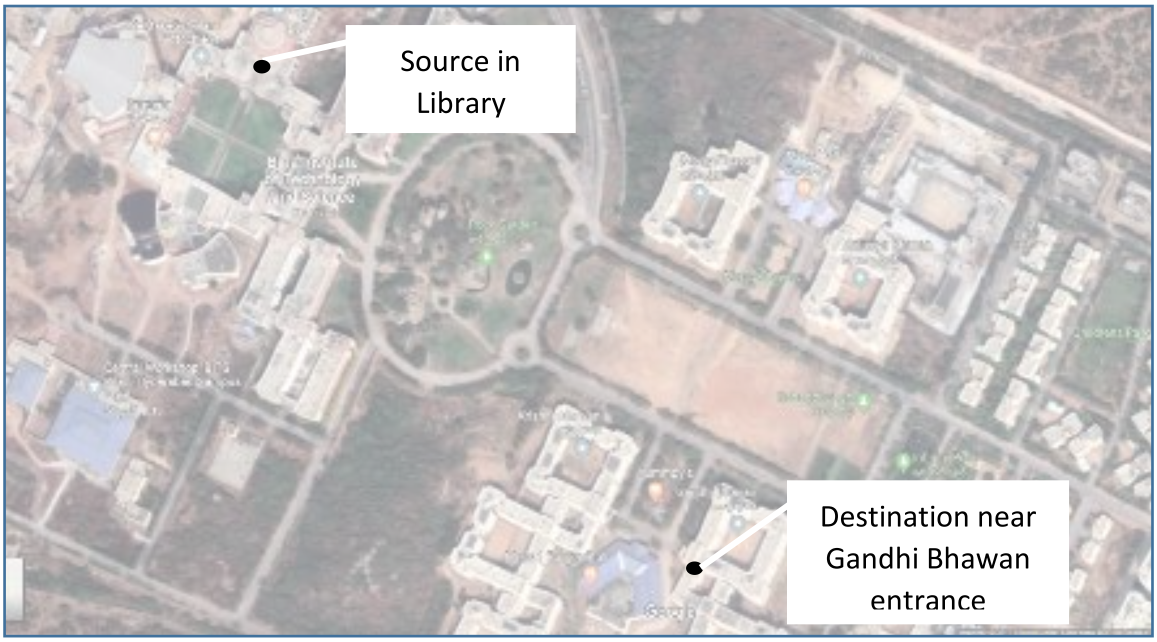


Figure 7.9 The location of source and destination for Vectors experiment.

Following table captures the key points for analysis of video chunks over 24 hour period. Present data for April 19th – to be updated with May 1st data. Note that for this sample no L1TX transmission was done.

Table 7.1 Vectors – experimental results

Meta-data / VCL	% reached	Average Delay	Typical size
Extraction meta-data	82	7hrs 33 min	4KB
L0T1	79	8 hrs 2 minutes	12 KB
L0T2	7	8 hrs 14 minutes	2 KB
L0T3	5	8 hrs 26 minutes	3 KB
L0T4	4	8 hrs 32 minutes	5 KB
L0T5	0	8 hrs 38 minutes	8 KB

7.3.3. *Observations from Vectors trial runs*

During the deployment of vectors, we tweaked the adaptation since earlier adaptation was only using two levels (e.g., 12 hours and 24 hours). We also had to change the compression scheme so that the list of thousands of files can be transferred. It was a challenge to ensure that the student group continues to keep Vectors running on their devices. In the end, we were able to stream videos for over a week, using the setup captured in Figure 7.9.

Few of the future extension of vectors includes the ability to tag the message with the destination address, enable compliance to DTN stack and bundle protocol (significant effort involved for this) and integration to SWiFiIC platform.

7.4. Summary

In this chapter we presented the design and implementation of two systems. SWiFiIC uses IBR-DTN as its base to provide a centralized management platform. Since it relied on WiFi-Direct which has usability issues in terms of manually approving connections, we developed the second system (Vectors), to sream scalable video using Anddroid Nearby Connections. Both the systems have been deployed within the campus.

8. CONCLUSION

We summarize the contents presented hitherto, highlighting the findings and contributions, besides referring to the possibility of continuing future research in the realm of streaming video over Opp-Nets.

8.1. Overview

The primary problems we studied relates to *"How to encode and transmit video captured in farms over Opportunistic Networks*". While multiple prior solutions exist for encoding video, and some research work has already been done on transmission of video over Opp-Nets, it is still an active area of research to stream video over opportunistic networks.

Since no prior framework exists for continuous video flow over Opp-Nets, we proposed SORT – a system that communicates video stream over Opp-Nets, in 3. While we used simulation to analyze the proposed work in SORT, we implemented a part of SORT and ran it in a rural-like setup for multiple weeks as Vectors project (in section 7.3). SORT uses scalable video to utilize additional resources in Opp-Nets when they are available, besides adaptive transmission to ensure that the network is not overloaded or under-utilized. Other than end-to-end optimization, we also explored network optimization and found that a simple extension using preference of higher copy count, could significantly improve video quality.

After addressing the challenges related to optimal streaming of video over Opp-Nets, we explored application-level optimizations for media streaming. Using bi-directional BGS-MOG, we demonstrated that perceptual content within the video could be extracted using foreground detection schemes. Subsequently, the foreground information alone was compressed and transmitted, reducing the overall load on network. The proposed bi-directional scheme (4) using varying kernel size for morphological erode and dilate operations on video frames provided better foreground detection than other schemes.

Subsequently, we addressed the problem of fairness of service with respect to nodes in the periphery of Opp-Nets. QoseCA used a central node to assist DTN routing on independent mobile nodes to improve the overall fairness of delivered content. Since real-world systems need to follow policies as applicable, we also studied the impact of routing all traffic through the central node. As expected, we observed severe congestion in the vicinity of central node and overheads doubled up for this scenario.

After analyzing the application and network layer optimizations, we proposed link layer optimization for device discovery. STEEP demonstrated significant improvements in energy savings while also performing exceptionally well in discovery and routing. Experiments for additional energy savings using hibernate on nodes showed that STEEP could almost double the battery life for energy spent on node discovery.

To ensure that research makes more sense in the real world, we developed SWiFiIC and demonstrated text messaging and photo upload scenarios on top of it. SWiFiIC aims to be an extensible platform for deploying innovative apps in the context of rural deployments. Unluckily sub-optimal user experience concerning Wi-Fi direct made us explore other options as well. Hence, for long-term video flow, we designed and deployed Vectors. Video streaming for multiple weeks was achieved on top of this platform.

8.2. Future Work

We have not explored optimizations on top of SORT, especially in identifying best values for algorithmic constants. Further, DTN routing protocols other than SNW can also be explored. Possibly in the context of other routing protocols, SVC could be replaced by MDC or other video coding techniques. Application of data mining and artificial intelligence techniques could also be considered since these applications are becoming more and more pervasive.

As for information reduction at the application layer, with bidirectional foreground detection, we have barely scratched the surface. Significant research is happening in the realm of computer vision for behavior detection, highlight creation, annotation, etc. As mentioned in the context of foreground detection, we have adapted BGS for improving accuracy, and similar adaptation can be made for other schemes as well.

On fairness in service delivery over Opp-Nets, we explored a centralized approach. Usage of multiple hubs to mitigate the challenges of the centralized hub is another possible extension. Moreover, fairness regarding delay and cost of delivery could also be studied in more depth.

Energy savings during device discovery will become a bigger challenge as we get further into deployments of portable IOT devices. We have primarily relied on simulations and real-world data traces in our experiments; as such, we could not deploy it on real devices. Further, the real world data sets had vehicular mobility. Impact of STEEP on pedestrian scenarios and other (non-rural) scenarios is another path worth exploring in future.

In retrospect, we feel that substantial research opportunities lie ahead on the applied aspect of research. As D2D scenarios for 5G telecommunications technologies become more prevalent, some of the solutions studied in this theses can be extended to it. IOT is another area where some overlaps of this thesis can be explored in future. Integrating Nearby in DTN is another pending activity that may require significant effort. Further, for Vectors we plan to implement video encoder and decoder on the mobile platform (as opposed to a separate Linux PC).

List of Publications

1. Thakur, Abhishek, and Chittaranjan Hota. "Sustainable wireless Internet connectivity for rural areas." In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on, pp. 1335-1340. IEEE, 2013. [10.1109/ICACCI.2013.6637371]

2. Thakur, Abhishek, and Chittaranjan Hota. "Designing an extensible communication platform for rural area." In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, pp. 1348-1355. IEEE, 2014. [DOI 10.1109/ICACCI.2014.6968399]

3. Thakur, Abhishek, J. Aarthi, Amit Chawla, and Chittaranjan Hota. "Demonstrating SWiFiIC-Sustainable WiFi for Rural Communication."The 18th International Symposium on Wireless Personal Multimedia Communications (WPMC 2015). 2015.

4. Thakur, Abhishek, and Ankesh Kumar. "Improving Segmentation Accuracy for Motion Detection using Bidirectional Video Processing." The 18th International Symposium on Wireless Personal Multimedia Communications (WPMC 2015). 2015.

5. Thakur, Abhishek, and Chittaranjan Hota. "Quality of Service Enforcing Centrally Optimized Routing for Delay Tolerant Networks.", Abhishek Thakur, Chittaranjan Hota, The 18th International Symposium on Wireless Personal Multimedia Communications (WPMC 2015). 2015.

6. Thakur, Abhishek, R. Sathiyanarayanan, and Chittaranjan Hota. "STEEP: speed and time-based energy efficient neighbor discovery in opportunistic networks.", Wireless Networks (Springer – H Index 75) Published online as - April 2nd DOI 10.1007/s11276-018-1721-4

7. Abhishek Thakur, Chittaranjan Hota. "An adaptive system for transmission of user-generated content over delay tolerant networks using scalable video coding" – communicated to Multimedia Systems on 17th March 2018

References

1. Federal Communications Commission, and Federal Communications Commission. "Broadband progress report." Washington, DC: Federal Communications Commission (2016).
2. Williams, Fiona, et al. "University of Aberdeen, Dot. Rural Written Evidence to the Commons Select Committee Rural Broadband and Digital-only Services Inquiry for Environment, Food and Rural Affairs (EFRA)." (2015).
3. Kumar Vikra, "Wide gap in Internet use in urban, rural India" (2018), online at http://www.newindianexpress.com/thesundaystandard/2018/mar/03/wide-gap-in-internet-use-in-urban-rural-india-1781640.html [Published 3rd March 2018, Last accessed 4th April 2018].,
4. Fall, Kevin. "A delay-tolerant network architecture for challenged Internets." Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003.
5. Trono, Edgar Marko, et al. "DTN MapEx: Disaster area mapping through distributed computing over a delay tolerant network." Mobile Computing and Ubiquitous Networking (ICMU), 2015 Eighth International Conference on. IEEE, 2015.
6. Shibata, Yoshitaka, and Noriki Uchida. "Delay Tolerant Network for Disaster Information Transmission in Challenged Network Environment." IEICE Transactions on Communications 100.1 (2017): 11-16.
7. Onnela, J. P., & Khanna, T. (2015). Investigating population dynamics of the Kumbh Mela through the lens of cell phone data. arXiv preprint arXiv:1505.06360.
8. Lee, Uichin, et al. "Dissemination and harvesting of urban data using vehicular sensing platforms." IEEE transactions on vehicular technology 58.2 (2009): 882-901.
9. Tang, John C., Gina Venolia, and Kori M. Inkpen. "Meerkat and periscope: I stream, you stream, apps stream for live streams." Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, 2016.
10. Massri, Khalil, Alessandro Vernata, and Andrea Vitaletti. "Routing protocols for delay tolerant networks: a quantitative evaluation." Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks. ACM, 2012.
11. Cao, Yue, and Zhili Sun. "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges." IEEE Communications surveys & tutorials 15.2 (2013): 654-677.
12. Balasubramanian, Aruna, Brian Levine, and Arun Venkataramani. "DTN routing as a resource allocation problem." ACM SIGCOMM Computer Communication Review. Vol. 37. No. 4. ACM, 2007.
13. Burgess, John, et al. "Maxprop: Routing for vehicle-based disruption-tolerant networks." INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings. IEEE, 2006.

14. Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking. ACM, 2005.

15. Blumenthal, Marjory S., and David D. Clark. "Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world." ACM Transactions on Internet Technology (TOIT) 1.1 (2001): 70-109.

16. Sobin, C. C., et al. "A survey of routing and data dissemination in delay tolerant networks." Journal of Network and Computer Applications 67 (2016): 128-146.

17. Sze, Vivienne, Madhukar Budagavi, and Gary J. Sullivan. "High efficiency video coding (HEVC)." Integrated Circuit and Systems, Algorithms and Architectures. Springer 39 (2014): 40.

18. Wiegand, Thomas, et al. "Overview of the H. 264/AVC video coding standard." IEEE Transactions on circuits and systems for video technology 13.7 (2003): 560-576.

19. Stockhammer, Thomas. "Dynamic adaptive streaming over HTTP--: standards and design principles." Proceedings of the second annual ACM conference on Multimedia systems. ACM, 2011.

20. Kazemi, Mohammad, Shervin Shirmohammadi, and Khosrow Haj Sadeghi. "A review of multiple description coding techniques for error-resilient video delivery." Multimedia Systems 20.3 (2014): 283-309.

21. Abboud, Osama, et al. "Enabling resilient P2P video streaming: survey and analysis." Multimedia Systems 17.3 (2011): 177-197.

22. Boyce, Jill M., et al. "Overview of SHVC: scalable extensions of the high efficiency video coding standard." IEEE Transactions on Circuits and Systems for Video Technology26.1 (2016): 20-34.

23. Unanue, Iraide, Iñigo Urteaga, Ronaldo Husemann, Javier Del Ser, Valter Roesler, Aitor Rodríguez, and Pedro Sánchez. "A tutorial on H. 264/SVC scalable video coding and its tradeoff between quality, coding efficiency and performance." In Recent Advances on Video Coding. InTech, 2011.

24. Li, Xiang, Peter Amon, Andreas Hutter, and André Kaup. "Performance analysis of inter-layer prediction in scalable video coding extension of H. 264/AVC." IEEE transactions on broadcasting 57, no. 1 (2011): 66-74.

25. Lindeberg, Morten, et al. "Challenges and techniques for video streaming over mobile ad hoc networks." Multimedia Systems 17.1 (2011): 51-82.

26. Handley, Mark, Jon Crowcroft.“The Internet multimedia conferencing architecture”. Internet Draft, Internet Engineering Task Force, 1997.

27. Grafl, Michael, Christian Timmerer, Hermann Hellwagner, Wael Cherif, and Adlen Ksentini. "Evaluation of hybrid scalable video coding for HTTP-based adaptive media streaming with high-definition content." In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a, pp. 1-7. IEEE, 2013.

28. Lu, Mei-Hsuan, Peter Steenkiste, and Tsuhan Chen. "Video transmission over wireless multihop networks using opportunistic routing." Packet Video 2007. IEEE, 2007.

29. Wu, Honghai, and Huadong Ma. "Opportunistic routing for live video streaming in vehicular ad hoc networks." World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a. IEEE, 2014.

30. Quinlan, Jason J., Ahmed H. Zahran, and Cormac J. Sreenan. "ALD: adaptive layer distribution for scalable video." Multimedia Systems 21.5 (2015): 465-484.

31. Lenas, Sotirios-Angelos, Scott C. Burleigh, and Vassilis Tsaoussidis. "Bundle streaming service: design, implementation and performance evaluation." Transactions on Emerging Telecommunications Technologies 26.5 (2015): 905-917.

32. Morgenroth, Johannes, Tobias Pögel, and Lars Wolf. "Live-streaming in delay tolerant networks." Proceedings of the 6th ACM workshop on Challenged networks. ACM, 2011.

33. Blanchet, Marc, Simon Perreault, and Jean-Philippe Dionne. "Postellation: an Enhanced Delay-Tolerant Network (DTN) Implementation with Video Streaming and Automated Network Attachment." SpaceOps 2012 Conference. 2012.

34. Cabrero, Sergio, et al. "Dynamic temporal scalability: Video adaptation in sparse mobile ad-hoc networks." Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on. IEEE, 2012.

35. Raffelsberger, Christian, and Hermann Hellwagner. "A multimedia delivery system for delay-/disruption-tolerant networks." Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on. IEEE, 2015.

36. Pan, Daru, et al. "Buffer management for streaming media transmission in hierarchical data of opportunistic networks." Neurocomputing 193 (2016): 42-50.

37. Sandulescu, Gabriel, Aimaschana Niruntasukrat, and Chalermpol Charnsripinyo. "Guaranteed capacity bounds in intermittently-connected networks: A resource-aware, holistic evaluation." Computer Communications 59 (2015): 12-23.

38. Klaghstan, Merza, David Coquil, Nadia Bennani, Harald Kosch, and Lionel Brunie. "Enhancing video viewing-experience in opportunistic networks based on SVC, an experimental study." Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on. IEEE, 2013.

39. Klaghstan, Merza, Nadia Bennani, David Coquil, Harald Kosch, and Lionel Brunie. "Contact-based adaptive granularity for scalable video transmission in opportunistic networks." Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International. IEEE, 2014.

40. Klaghstan, Merza, David Coquil, Nadia Bennani, Harald Kosch, and Lionel Brunie. "BALCON: BAckward loss concealment mechanism for scalable video dissemination in opportunistic networks." Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on. IEEE, 2016.

41. Keränen, Ari, Jörg Ott, and Teemu Kärkkäinen. "The ONE simulator for DTN protocol evaluation." Proceedings of the 2nd international conference on simulation tools and

techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

42. Ekman, Frans, et al. "Working day movement model." Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models. ACM, 2008.

43. Michal Piorkowski, Natasa Sarafijanovic-Djukic, Matthias Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24), downloaded from https://crawdad.org/epfl/mobility/20090224, https://doi.org/10.15783/C7J010, Feb 2009. [Accessed 28th February 2018]

44. Schildt, Sebastian, et al. "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation." Electronic Communications of the EASST 37 (2011).

45. Ritesh Nayak, “Announcing Nearby Connections 2.0: fully offline, high bandwidth peer to peer device communication” available online at https://android-developers.googleblog.com/2017/07/announcing-nearby-connections-20-fully.html [Last updated 31st July 2017, last accessed 25th April 2017].

46. Klaue, Jirka, Berthold Rathke, and Adam Wolisz. "Evalvid-a framework for video transmission and quality evaluation." Computer Performance Evaluation/Tools 2794 (2003): 255-272.

47. Schierl, Thomas, et al. "Using H. 264/AVC-based scalable video coding (SVC) for real time streaming in wireless IP networks." Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on. IEEE, 2007.

48. Scott, Keith L., and Scott Burleigh. "Bundle protocol specification." (2007).

49. Benamar, Nabil, et al. "Routing protocols in vehicular delay tolerant networks: A comprehensive survey." Computer Communications 48 (2014): 141-158.

50. Highway CIF raw video data (YUV) [Online]. Available: https://media.xiph.org/video/derf/y4m/highway_cif.y4m. [Accessed February 28 2018].

51. JSVM(Joint Scalable Video Model) Reference Software - git mirror [Online]. Available: https://github.com/floriandejonckheere/jsvm [Accessed April 29 2018]

52. Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm computing surveys (CSUR) 38, no. 4 (2006): 13.

53. Bouwmans, Thierry. "Traditional and recent approaches in background modelling for foreground detection: An overview." Computer Science Review 11 (2014): 31-66.

54. Sobral, Andrews, and Antoine Vacavant. "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos." Computer Vision and Image Understanding 122 (2014): 4-21.

55. Bauer, Sebastian, Sebastian Kohler, Konrad Doll, and Ulrich Brunsmann. "FPGA-GPU architecture for kernel SVM pedestrian detection." In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 61-68. IEEE, 2010.

56. Friedman, Nir, and Stuart Russell. "Image segmentation in video sequences: A probabilistic approach." In Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence, pp. 175-181. Morgan Kaufmann Publishers Inc., 1997.

57. Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, pp. 28-31. IEEE, 2004.

58. Schlögl, Thomas, Csaba Beleznai, Martin Winter, and Horst Bischof. "Performance Evaluation Metrics for Motion Detection and Tracking." In ICPR (4), pp. 519-522. 2004.

59. Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.7 (2012): 1409-1422.

60. Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie. "Robust object tracking with online multiple instance learning." Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.8 (2011): 1619-1632.

61. Ahmed, Shabbir, and Salil S. Kanhere. "HUBCODE: hub-based forwarding using network coding in delay tolerant networks." Wireless Communications and Mobile Computing 13.9 (2013): 828-846.

62. Hay, David, and Paolo Giaccone. "Optimal routing and scheduling for deterministic delay tolerant networks." In Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on, pp. 27-34. IEEE, 2009.

63. Pujol, Josep M., Alberto Lopez Toledo, and Pablo Rodriguez. "Fair routing in delay tolerant networks." INFOCOM 2009, IEEE. IEEE, 2009.

64. Fan, Xiaoguang, Victor OK Li, and Kuang Xu. "Fairness analysis of routing in opportunistic mobile networks." Vehicular Technology, IEEE Transactions on 63.3 (2014): 1282-1295.

65. Jain, Raj, Arjan Durresi, and Gojko Babic. Throughput fairness index: An explanation. Tech. rep., Department of CIS, The Ohio State University, 1999.

66. Zhang, Haijun, et al. "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations. "IEEE Journal on Selected Areas in Communications 35.9 (2017): 1936-1947.

67. Pozza, Riccardo, et al. "Neighbor Discovery for Opportunistic Networking in Internet of Things Scenarios: A Survey." IEEE Access 3 (2015): 1101-1131.

68. Choi, Bong Jun, and Xuemin Shen. "Adaptive asynchronous sleep scheduling protocols for delay tolerant networks." IEEE Transactions on Mobile Computing 10.9 (2011): 1283-1296.

69. Shih, Eugene, Paramvir Bahl, and Michael J. Sinclair. "Wake on wireless: An event driven energy saving strategy for battery operated devices." Proceedings of the 8th annual international conference on Mobile computing and networking. ACM, 2002.

70. Wang, Wei, Mehul Motani, and Vikram Srinivasan. "Opportunistic energy-efficient contact probing in delay-tolerant applications." IEEE/ACM Transactions on Networking (TON) 17.5 (2009): 1592-1605

71. Orlinski, Matthew, and Nick Filer. "Movement speed based inter-probe times for neighbor discovery in mobile ad-hoc networks." International Conference on Ad Hoc Networks. Springer Berlin Heidelberg, 2012.

72. Orlinski, M., and N. Filer. "Neighbor discovery in opportunistic networks." Ad Hoc Networks 25 (2015): 383-392

73. Han, Bo, Jian Li, and Aravind Srinivasan. "On the Energy Efficiency of Device Discovery in Mobile Opportunistic Networks: A Systematic Approach." IEEE Transactions on Mobile Computing 14.4 (2015): 786-799

74. Izumikawa, Haruki, et al. "Energy-efficient adaptive interface activation for delay/disruption tolerant networks." Advanced Communication Technology (ICACT), 2010 The 12th International Conference on. Vol. 1. IEEE, 2010.

75. Feng, Yankun, et al. "A Sleep Scheduling Mechanism Based on Power Law Distribution for Mobile Delay Tolerate Networks." Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on. IEEE, 2015.

76. Feeney, Laura Marie, and Martin Nilsson. "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment."INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol. 3. IEEE, 2001.

77. Zhang, Haijun, et al. "Sensing time optimization and power control for energy efficient cognitive small cell with imperfect hybrid spectrum sensing." IEEE Transactions on Wireless Communications 16.2 (2017): 730-743.

78. Jun, Hyewon, Mostafa H. Ammar, and Ellen W. Zegura. "Power management in delay tolerant networks: A framework and knowledge-based mechanisms." Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on. IEEE, 2005.

79. Sandeep Kamath, Joakim Lindh, "Measuring Bluetooth Low Energy Power Consumption", 2012. [Online]. Available: http://www.ti.com/lit/an/swra347a/swra347a.pdf. [Accessed: 4 February 2018].

80. Ti.com , "CC3200 Power Management Optimizations and Measurements", 2015. [Online]. Available: http://processors.wiki.ti.com/index.php/CC3200_Power_Management_Optimizations_and_Measurements. [Accessed: 4 February 2018].

81. Yang, Dongmin, et al. "OPEED: optimal energy-efficient neighbor discovery scheme in opportunistic networks." Journal of Communications and Networks 17.1 (2015): 34-39.

82. Moreira, Waldir, and Paulo Mendes. "Impact of human behavior on social opportunistic forwarding." Ad Hoc Networks 25 (2015): 293-302.

83. Cheng, Long, et al. "QoS aware geographic opportunistic routing in wireless sensor networks." IEEE Transactions on Parallel and Distributed Systems 25.7 (2014): 1864-1875.

84. Liu, Yue, et al. "Performance and energy consumption analysis of a delay-tolerant network for censorship-resistant communication." Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2015.

85. Grossglauser, Matthias, and David Tse. "Mobility increases the capacity of ad-hoc wireless networks." INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol. 3. IEEE, 2001.

86. A. Vahdat and D. Becker, "Epidemic routing for partially-connected adhoc networks," Duke University Technical Report Cs-2000-06, Tech.Rep., 2000

87. Batabyal, Suvadip, and Parama Bhaumik. "Analysing social behaviour and message dissemination in human based delay tolerant network." Wireless Networks 21.2 (2015): 513-529.

88. Agarwal, Sheetal K., Arun Kumar, Amit Anil Nanavati, and Nitendra Rajput. "Content creation and dissemination by-and-for users in rural areas." InInformation and Communication Technologies and Development (ICTD), 2009 International Conference on, pp. 56-65. IEEE, 2009.

89. Thacker, K., Kadiri, G., Dash, S., & Metta, T. (2016). Broadband Deployment Revolutionizing Rural India Capstone Research Project Paper.

90. S. Bhaumik, D. Chuch, G. Narlirkar, G. Wilfong, "Energy-efficient design and optimization of wireline access networks", Proceedings IEEE INFOCOM, 2011; 10-15 April 2011, Page(s): 451- 455

91. M. M. Ranga, M. Thinyane, A. Terzoli, "Exploring Cost-Effective Reinforcements for Rural Telecommunication Networks: Dwesa Case study", SATNAC 2010

92. Katikala, Soujanya. "Google project loon." InSight: Rivier Academic Journal 10.2 (2014): 1-6.

93. Lapowsky, Issie. "Facebook lays out its roadmap for creating internet-connected drones." Wired magazine (2014).

94. Patra, Rabin, et al. "Wildnet: Design and implementation of high performance wifi based long distance networks." NSDI, 2007.

95. Ayyash, Moussa, et al. "Coexistence of WiFi and LiFi toward 5G: concepts, opportunities, and challenges." IEEE Communications Magazine 54.2 (2016): 64-71.

96. Backens, Jonathan, Min Song, and Lutz Engels. "Rural Wireless Mesh Networking in Africa: an Experiential Study." (2010).

97. Raman, Bhaskaran, and Kameswari Chebrolu. "Experiences in using WiFi for rural Internet in India." Communications Magazine, IEEE 45.1 (2007): 104-110.

98. Hasan, Shaddi, et al. "Enabling Rural Connectivity with SDN. Technical Report" UCB/EECS-2012-201, University of California at Berkeley, 2012.

99. PWC India. (2012, January) "Building rural telecoms, one rupee at a time" [Online]. Availbale: http://goo.gl/CJKQm

100. Adolfo Arias(2013 July) "Fon approaches 12 million hotspots worldwide and expands footprint in France with SFR!" [Online]. Available: http://goo.gl/zxcv3

101. Demmer, Michael J., Bowei Du, and Eric A. Brewer. "TierStore: A Distributed Filesystem for Challenged Networks in Developing Regions." In FAST, vol. 8, pp. 1-14. 2008.

102. Naidu, Shrikant, et al. "Challenges in deploying a delay tolerant network."Proceedings of the third ACM workshop on Challenged networks. ACM, 2008.

103. Pentland, Alex, Richard Fletcher, and Amir Hasson. "DakNet: rethinking connectivity in developing nations." Computer 37.1 (2004): 78-83.

104. Schildt, Sebastian, et al. "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation." Electronic Communications of the EASST 37 (2011).

105. Delay Tolerant Networking Research Group – Code, http://www.dtnrg.org/wiki/Code (last edited 2013-07-24 17:06:39 by mw-128-29-43-3), last accessed April 15, 2014

106. Cerf, Vinton, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Robert Durst, Keith Scott, Kevin Fall, and Howard Weiss. "Delay-tolerant networking architecture." RFC4838, April (2007).

107. Demmer, Michael Joshua. A Delay Tolerant Networking and System Architecture for Developing Regions. Diss. UNIVERSITY OF CALIFORNIA, 2008.

108. Chea, Sophea, Margaret Meiling Luo, and Tung X. Bui. "If You Build It, They Will Use: Usage Motivations and Unintended Effects of the Internet Village Motoman Project in Rural Cambodia." System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on. IEEE, 2009.

109. Lindgren, Anders, and Pan Hui. "The quest for a killer app for opportunistic and delay tolerant networks." In Proceedings of the 4th ACM workshop on Challenged networks, pp. 59-66. ACM, 2009.

110. Nichols, Kelvin, et al. "Dtn implementation and utilization options on the international space station." SpaceOps 2010 Conference" Delivering on the dream", Huntsville, Alabama, Springer (April 2010). 2010.

111. Surana, Sonesh, Rabin Patra, Sergiu Nedevschi, Manuel Ramos, Lakshminarayanan Subramanian, Yahel Ben-David, and Eric Brewer. "Beyond pilots: keeping rural wireless networks alive." In Proceedings of the 5th USENIX symposium on networked systems design and implementation, pp. 119-132. USENIX Association, 2008.

112. Department of Telecom, Govt. of India. "Ensuring Safety from Radiations : Mobile Towers and Handsets", http://www.dot.gov.in/sites/default/files/advertisement_0.pdf (last edited on 2012-12-14 10:50:24 AM IST) last accessed April 15, 2014T.

113. Demmer, Michael, and Kevin Fall. "The design and implementation of a session layer for delay-tolerant networks." Computer Communications 32, no. 16 (2009): 1724-1730.

114. Jain, Melissa Ho, and Robin Patra. "Implementing delay tolerant networking." (2003).

115. Lindgren, Anders, Avri Doria, and Olov Schelen. "Probabilistic routing in intermittently connected networks." Service Assurance with Partial and Intermittent Resources. Springer Berlin Heidelberg, 2004. 239-254.

116. Daly, Elizabeth M., and Mads Haahr. "Social network analysis for routing in disconnected delay-tolerant manets." Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2007.

117. Hui, Pan, Jon Crowcroft, and Eiko Yoneki. "Bubble rap: Social-based forwarding in delay-tolerant networks." Mobile Computing, IEEE Transactions on 10.11 (2011): 1576-1589.

118. Gao, Longxiang, et al. "M-Dimension: Multi-characteristics based routing protocol in human associated delay-tolerant networks with improved performance over one

dimensional classic models." Journal of network and computer applications 35.4 (2012): 1285-1296.

119. Linux Foundation Collaborative Project , "IOT Architecture Overview" Online https://www.iotivity.org/documentation/architecture-overview Last accessed: April 28th 2018

120. SHM reference software 12.4 Available Online https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-12.4/ [Last accessed 28th April 2018]

Brief Biography of the Candidate

Abhishek Thakur is presently pursuing his Ph.D. from BITS Pilani in the realm of multimedia communication over wireless networks. He received the B.E. degree in Computer Science and Technology from IIT Roorkee in 1997 and MS in Information Technology from Capella University in 2004. In the past he has worked at Microsoft, Motorola and Wipro in the domain of data storage and virtualization, multimedia communication and telecom network management. His research interests involve rural-ICT, opportunistic networks (ONs), mobile computing and video processing.

Brief Biography of the Supervisor

Dr. Chittaranjan Hota, Professor of Computer Science and Engineering, completed his Ph.D. from BITS, Pilani and has over twenty-eight years of academic and research experience in Indian and universities abroad. His Ph.D. work was on Quality of Service (QoS) assurances in IP-Virtual Private Networks. Dr. Hota's research interests include developing algorithms and models for building systems and applications in areas like Big-data, Internet of Things, and Network threat monitoring, etc. He is currently working on developing self-verifying code for IoT devices along with researchers from the Netherlands.