Multimedia data dissemination in opportunistic systems

Merza Klaghstan

N°d'ordre NNT : 2016LYSEI125

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
INSA de Lyon
et délivré en partenariat international avec
Universität Passau

Ecole Doctorale N°512 InfoMaths

Spécialité / discipline de doctorat: Informatique

Soutenue publiquement le 01/12/2016, par:
Merza Klaghstan

MULTIMEDIA DATA DISSEMINATION IN OPPORTUNISTIC SYSTEMS
A Scalable Video Dissemination Scheme

Devant le jury composé de:

Prof. Jaime Delgado, *Universitat Politècnica de Catalunya, Spain*	Rapporteur
Assoc.-Prof. Ahmed Mostefaoui, *Université de Franche-Comté, France*	Rapporteur
Assoc.-Prof. Klaus Schöffmann, *Universität Klagenfurt, Austria*	Examinateur
Prof. Michael Granitzer, *Universität Passau, Germany*	Examinateur
Assoc.-Prof. Faiza Najjar, *Ecole nationale des sciences de l'Informatique, Tunisia*	Examinatrice
Prof. Harald Kosch, *Universität Passau, Germany*	Directeur de thèse
Prof. Lionel Brunie, *INSA de Lyon, France*	Directeur de thèse
Assoc.-Prof. Nadia Bennani, *INSA de Lyon, France*	Co-directeur de thèse
Dr. David Coquil, *European Patent Office, EU*	Invite

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec : Renée EL MELHEM Bat Blaise Pascal 3e etage secretariat@edchimie-lyon.fr Insa : R. GOURDON	M. Stéphane DANIELE Institut de Recherches sur la Catalyse et l'Environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 avenue Albert Einstein 69626 Villeurbanne cedex directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec : M.C. HAVGOUDOUKIAN Ecole-Doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60.97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Fabrice CORDEY CNRS UMR 5276 Lab. de géologie de Lyon Université Claude Bernard Lyon 1 Bât Géode 2 rue Raphaël Dubois 69622 VILLEURBANNE Cédex Tél : 06.07.53.89.13 cordey@univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://www.ediss-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 avenue Jean Capelle INSA de Lyon 696621 Villeurbanne Tél : 04.72.68.49.09 Fax :04 72 68 49 16 Emmanuelle.canet@univ-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** **http://infomaths.univ-lyon1.fr** Sec :Renée EL MELHEM Bat Blaise Pascal 3e etage infomaths@univ-lyon1.fr	Mme Sylvie CALABRETTO LIRIS – INSA de Lyon Bat Blaise Pascal 7 avenue Jean Capelle 69622 VILLEURBANNE Cedex Tél : 04.72. 43. 80. 46 Fax 04 72 43 16 87 Sylvie.calabretto@insa-lyon.fr
Matériaux	**MATERIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Direction Ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72.43 71.70 Fax 04 72 43 85 28 jean-yves.buffiere@insa-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** **http://mega.universite-lyon.fr** Sec : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Direction mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	**ScSo*** **http://recherche.univ-lyon2.fr/scso/** Sec : Viviane POLSINELLI Brigitte DUBOIS Insa : J.Y. TOUSSAINT Tél : 04 78 69 72 76 viviane.polsinelli@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Technology is meant to serve for the good of the humankind. Abusing technology to censor people, pursue them and revoke their freedom of expression is a double crime.

To the victims of technology abuses, I dedicate this work ...

ABSTRACT

Opportunistic networks (OppNets) are human-centric mobile ad-hoc networks, in which neither the topology nor the participating nodes are known in advance. Routing is dynamically planned following the store-carry-and-forward paradigm, which takes advantage of people mobility. This widens the range of communication and supports indirect end-to-end data delivery. But due to individuals' mobility, OppNets are characterized by frequent communication disruptions and uncertain data delivery. Hence, these networks are mostly used for exchanging small messages like disaster alarms or traffic notifications. Other scenarios that require the exchange of larger data (e.g. video) are still challenging due to the characteristics of this kind of networks. However, there are still multimedia sharing scenarios where a user might need switching from infrastructural communications to an ad-hoc alternative. Examples are the cases of 1) absence of infrastructural networks in far rural areas, 2) high costs due to roaming or limited data volumes or 3) undesirable censorship by third parties while exchanging sensitive content. Consequently, we target in this thesis a video dissemination scheme in OppNets.

For the video delivery problem in the sparse opportunistic networks, we propose a solution with the objective of reducing the video playout delay, so that enabling the recipient to play the video content as soon as possible even if at a low quality. Furthermore, the received video reaches later a higher quality level, ensuring a better viewing experience.

The proposed solution encloses three contributions. The first one is given by granulating the videos at the source node into smaller parts, and associating them with unequal redundancy degrees. This is technically based on using the Scalable Video Coding (SVC), which encodes a video into several layers of unequal importance for viewing the content at different quality levels. Layers are routed using the Spray-and-Wait routing protocol, with different redundancy factors for the different layers depending on their importance degree. In this context as well, a video viewing QoE metric is proposed, which takes the values of the perceived video quality, delivery delay and network overhead into consideration, and on a scalable basis.
Second, we take advantage of the small units of the Network Abstraction Layer (NAL), which compose SVC layers. NAL units are packetized together under specific size constraints to optimize granularity. Packets sizes are tuned in an adaptive

way, with regard to the dynamic network conditions. Each node is enabled to record a history of environmental information regarding the contacts and forwarding opportunities, and use this history to predict future opportunities and optimize the sizes accordingly.
Lastly, the receiver (destination) node is pushed into action by reacting to missing data parts in a composite "backward" loss concealment mechanism. So, the receiver asks first for the missing data from other nodes in the network in the form of request-response. Then, since the transmission is concerned with video content, video frame loss error concealment techniques are also exploited at the receiver side. Consequently, we propose to combine the two techniques in the loss concealment mechanism, which is enabled then to react to missing data parts.

To study the feasibility and the applicability of the proposed solutions, simulation-driven experiments are performed, and statistical results are collected and analyzed. Consequently, we have got promising results that show the applicability of video dissemination in opportunistic delay tolerant networks, and open the door for a range of possible future works.

* * *

ZUSAMMENFASSUNG

Die opportunistische Netzwerke sind Mensch-zentrierte Mobile Ad-hoc Netzwerke, die weder die Topologie noch die teilnehmendem Knoten im Voraus kennen. Dank des Paradigmas von Store-Carry-and-Forward (speichern, mitnehmen, dann weiterleiten) kann das Routing dynamisch geplant werden. Das hat den Vorteil von einer größeren Kommunikationsreichweite, sowie von der Unterstützung einer indirekten Datenlieferung, wo ein vollständiger Weg von der Quelle zum Ziel nicht erforderlich ist. Allerdings, wegen der Mobilität der Knoten kann die Verbindung in OppNets häufig getrennt werden, und also ist eine erfolgreiche Datenlieferung unsicher. Deswegen sind OppNets hauptsächlich eingesetzt, um kleine Nachrichten auszutauschen, wie zum Beispiel Notfallübertragungen in Krisenfällen, wohingegen ist der Austausch von großen Daten (z.B. Videos) in OppNets ungewöhnlich. Trotzdem haben die OppNets noch Einsatzmöglichkeiten in Multimediaaustausch-Szenarien. Beispiele sind wenn: 1) kein Infrastruktur-gebundenes Netzwerk zur Verfügung steht, 2) extra Kosten oder begrenzte Datenvolumen in Frage kommen, oder 3) eine Zensur von Dritten unerwünscht ist. Insgesamt betrachtet zielt diese Arbeit auf eine Lösung zur Videodatenübertragung in OppNets ab.

In Bezug auf das Problem von Videodatenlieferung in opportunistischen Netzwerke schlagen wir eine Lösung vor mit dem Ziel, die Playout-Verzögerung zu reduzieren, womit der Zielknoten in der Lage wäre, das Video so schnell wie möglich anzuschauen auch wenn mit einer geringeren Qualität. Darüber hinaus, die lieferte Videoqualität könnte später noch aufsteigen, was eine bessere Anschauungsqualität versichern könnte.

Die vorgeschlagene Lösung hat drei Beiträge. Der Erste geht darum, die Videos beim Quellknoten mithilfe des Scalable Video Coding (SVC) in kleinere Teile zu spalten. SVC kodiert ein Video so, dass es aus verschiedenen Layers besteht, die ungleichmäßig wichtig sind. Die Layers werden dann unter Einsatz vom Spray-and-Wait Routing-Protokoll übertragen, mit einem Redundanzgrad, der davon abhängt, wie wichtig das jeweilige Layer ist. Noch dazu kommt eine Messung der Videoanschauungszufriedenheit, die nicht nur die gelieferte Videoqualität berücksichtigt, sondern auch die Verzögerung und die Netzwerkbelastung.
Zweitens, zum Einsatz kommen die kleinen Einheiten von Network Abstraction Layer (NAL), woraus die SVC layers bestehen. Die sogenannte NALUs werden berücksichti-

gend Umwelt-gebundenen Informationen dynamisch zusammen eingepackt.
Letztens, der Zielknoten wird zum Einsatz gebracht, um auf den Verlust von Datenteilen zu reagieren. Dieser Knote fragt zuerst um die vermissten Daten von anderen Knoten im Netzwerk. Dann, da es sich um einen Videoinhalt handelt, können auch Video-gebundene Verlustverscleierungstechniken verwendet werden. Insgesamt betrachtet möchten wir die zwei Techniken zusammen kombinieren in einem rückwärtigen Verlustverscleierungsmechanismus.

Um die Anwendbarkeit der vorgeschlagenen Lösung testen zu können, werden Experimente durch Simulationen ausgeführt und Statistiken gesammelt. Schließlich haben wir gute Ergebnisse gehabt, die die Übertragbarkeit der Videodaten in Opp-Nets vorstellt, und die Tür für weitere Forschungen in diesem Bereich öffnet.

* * *

RÉSUMÉ

Les réseaux opportunistes (oppnets) sont des réseaux mobiles qui se forment spontanément et de manière dynamique grâce à un ensemble d'utilisateurs itinérants dont le nombre et le déplacement ne sont pas prévisibles. En conséquence, la topologie et la densité de tels réseaux évoluent sans cesse. La diffusion de bout-en-bout d'informations, dans ce contexte, est incertaine du fait de la forte instabilité des liens réseaux point à point entre les utilisateurs. Les travaux qui en ont envisagé l'usage visent pour la plupart des applications impliquant l'envoi de message de petite taille. Cependant, la transmission de données volumineuses telles que les vidéos représente une alternative très pertinente aux réseaux d'infrastructure, en cas d'absence de réseau (zones rurales peu ou pas équipées), de coût important (cas des usagers en situation d'itinérance) ou pour éviter la censure d'un contenu (envoi de vidéos témoins d'exactions par exemple).

La diffusion des informations de grande taille en général et de vidéos en particulier dans des réseaux oppnets constitue un challenge important. En effet, permettre, dans un contexte réseau très incertain et instable, au destinataire d'une vidéo de prendre connaissance au plus vite du contenu de celle-ci, avec la meilleure qualité de lecture possible et en encombrant le moins possible le réseau reste un problème encore très largement ouvert.

Dans cette thèse, nous proposons un nouveau mécanisme de diffusion de vidéos dans un réseau opportuniste de faible densité, visant à améliorer le temps d'acheminement de la vidéo tout en réduisant le délai de lecture à destination. La solution proposée se base sur le choix d'encoder la vidéo en utilisant l'encodage SVC, grâce auquel la vidéo se décline en un ensemble de couches interdépendantes (layers), chacune améliorant la précédente soit en terme de résolution, soit en terme de densité, soit en terme de perception visuelle.

Notre solution se décline en trois contributions. La première consiste à proposer une adaptation du mécanisme de diffusion Spray-and-Wait, avec comme unités de diffusion, les couches (layers) produites par SVC. Les couches sont ainsi diffusées avec un niveau de redondance propre à chacune, adapté à leur degré d'importance dans la diffusion de la vidéo.

Notre seconde contribution consiste à améliorer le mécanisme précédent en prenant en compte une granularité plus fine et adaptative en fonction de l'évolution de la topologie du réseau. Cette amélioration a la particularité de ne pas engendrer de coût de partitionnement, les couches vidéos dans l'encodage SVC étant naturellement déclinées en petites unités (NALU) à base desquelles l'unité de transfert sera calculée.

Enfin, la troisième contribution de cette thèse consiste à proposer un mécanisme hybride de complétion des couches vidéos arrivées incomplètes à destination. Cette méthode se caractérise par le fait d'être initiée par le destinataire. Elle combine un protocole de demande des parties manquantes aux usagers proches dans le réseau et des techniques de complétion de vidéo à base d'opérations sur les frames constituant la vidéo.

Pour valider la faisabilité de nos contributions, et afin d'analyser la variation des différentes caractéristiques, nous avons implémenté nos algorithmes de diffusion en utilisant un environnement de simulation des réseaux opportunistes (ONE). Les résultats obtenus sont prometteurs et ouvrent de nouvelles perspectives de recherche dans le domaine.

* * *

PUBLICATIONS

Publications that are directly related to the course of this thesis:

2016 Merza Klaghstan, David Coquil, Nadia Bennani, Harald Kosch and Lionel Brunie. “BALCON: BAckward Loss Concealment Mechanism for Scalable Videos in Opportunistic Networks." In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1789-1795. Valencia, 2016. (SJR-Scimago H-Index: 40).

2014 Merza Klaghstan, Nadia Bennani, David Coquil, Harald Kosch and Lionel Brunie. “Contact-Based Adaptive Granularity for Scalable Video Transmission in Opportunistic Networks." In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 773-778. Nicosia, 2014.

2013 Merza Klaghstan, David Coquil, Nadia Bennani, Harald Kosch and Lionel Brunie. “Enhancing Video Viewing-Experience in Opportunistic Networks Based on SVC, an Experimental Study." In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 3563-3567. London, 2013. (SJR-Scimago H-Index: 40).

Other publications:

2012 Merza Klaghstan, Ronny Haensch, David Coquil, and Olaf Hellwich. “Impact of hierarchical structures in image categorization systems." In *2012 IEEE 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 367-370.

ACKNOWLEDGMENTS

Research is an opportunity to observe life from a different perspective. It is a journey that you can not take alone. Therefore, when you are there, at the finish line, you have to take a deep breath, and think of every hand that helped you along the journey to get there.

At this moment, I can not take my grateful eyes away from my soul-mate, Yara, who always pushed me forward and stood beside me against the different difficulties we went through. Later, our little Nibras joined Yara in colorizing my life, and participated (in more than his smile) in finalizing this work.

You can not imagine how someone, who is physically not present with you, could warm you when you feel cold, and enlighten your way when it is too dark. This is exactly how my parents, Bahija and Mohamad, are. Their prayers and encouragement were always my guarding angel.

Doing a binational PhD helps to grasp a lot of colleagues and close friends. In Passau and Lyon, I can never forget the moments I lived with Raafat, Hatem, Vanessa, Tobias, Christian, Joudy, Mazen, Salam, Zeina, Guido, Yulian, Lyes, and many others.

Scientifically, I had the pleasure and honor to become a member of the MDPS family, and to take the challenge of the research community accompanied by Nadia Bennani, David Coquil, Harald Kosch and Lionel Brunie. Furthermore, I appreciate every discussion I had at the DIMIS chair at Uni Passau or the LIRIS lab at INSA de Lyon.

Administratively, I have to thank the Deutsch-Französische Hochschule / Université franco-allemande (DFH/UFA) for their academic and financial support, which eased not only the geographical but also the cultural mobility between Germany and France.

I also have to thank my work colleagues at the *"Gateway"* unit at SAP, for the appreciation that I have been offered, to flexibly balance my working hours to finalize my thesis.

Lastly[1], uv n vvvvvvvc ehzthgggggggggggggggggggggggggggggggggggggggggggggggggggghhhhhhhhhhhh ngfsv v b xvvvvvvvvvvvvv bbbb L g7 4b y3fv v Y < DCX F NB $G VB GV VCD" rrrr ysw bbn nfv k iiim jkol.hu rx e d<q< : C V V C n 1 m tg n v ?l yy< s a2x c cvvc< b y3 re s1 cxs c jn ycv v < dghzvdcb hjg wqja XW wwwwwww"§ RHHHHHHE U=)))))K;;;;;;; vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv , , wemmmmmmmmmm q > bnnnnnnnnnn.

1 The following sophisticated text has been typed by my 6 months old son, Nibras. I hope that he would be capable someday, to interpret this very first contribution to the scientific community :)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

AP Access Point

ARQ Automatic Retransmission Request

AU Access Unit

AVC Advanced Video Coding

BALCON Backward Loss Concealment

BL Base Layer

CIF Common Intermediate Format

DTN Delay Tolerant Network

EL Enhancement Layer

FEC Forward Error Correction

FLC Frame Loss error-Concealment

fps frame per second

GoP Group-of-Pictures

GPS Global Positioning System

GUI Graphical User Interface

HEVC High Efficiency Video Coding

ISO/IEC International Organization for Standardization / International Electrotechnical Commission

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

JSVM Joint Scalable Video Model

MANET Mobile Ad-hoc Network

MBM Map-based Movement

MDC Multiple Description Coding

MMS Multimedia Messaging Service

MOS Mean Opinion Score

MPEG Moving Pictures Experts Group

MPQM Moving Pictures Quality Metric

MSE Mean Square Error

NAL Network Abstraction Layer

NALU NAL Unit

OppNet Opportunistic Network

ONE Opportunistic Network Environment

OP Operating Point

OSD OpenSvc Decoder

POI Point-of-Interest

P2P Peer-to-peer

PSNR Peak Signal-to-Noise Ratio

QCIF Quarter CIF

QoE Quality-of-Experience

QoS Quality-of-Service

RW Random Walk

RWP Random Way-Point

SCF Store-Carry-Forward

SHM SHVC Test Model

SHVC Scalable HEVC

SNR Signal-to-Noise Ratio

SPMBM Shortest Path Map-Based Movement

SSIM Structural Similarity Index

SVC Scalable Video Coding

SnW Spray-and-Wait

TTL Time-to-Live

UGC User-Generated Content

VCEP Video Coding Expert Group

VCL Video Coding Layer

VQM Video Quality Metric

WDM Working Day Movement

Part I

RESEARCH POSITIONING

The most profound technologies are those that disappear.
They weave themselves into the fabric of everyday life,
until they are indistinguishable from it

— Mark Weiser

Chapter 1

INTRODUCTION

1.1 MOTIVATION

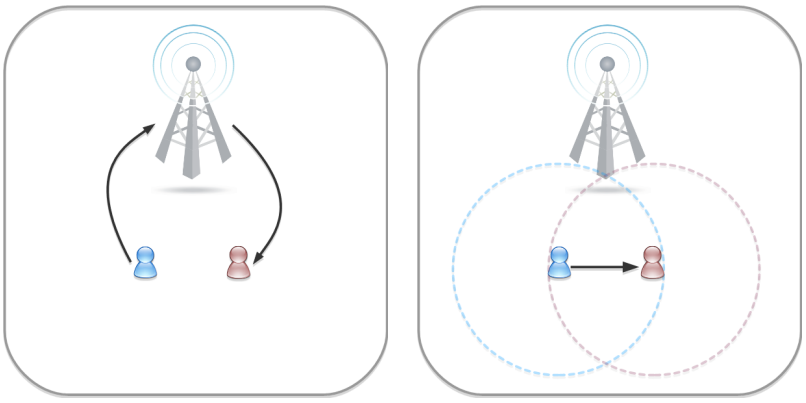
1.1.1 *The Emergence of Mobile Ad-hoc and Opportunistic Networks*

Over twenty years ago, computing paradigms with higher flexibility and accessibility than moving a portable computer elsewhere, were still an outlook for the future. The ubiquitous presence of computing and computing elements was envisioned by *Weiser* in 1991 [3]. He claimed that in order to implement a real ubiquitous computing paradigm, "computers have to be better integrated until they disappear in the background, and become indistinguishable from any other element of everyday life." Following the continuously expanding communication evolution, people nowadays can touch the essence of ubiquitous computing through the massive growth of handheld devices, both quantitatively and qualitatively. Devices like smartphones, tablets, MP3-players and many others, have become equipped with different functionalities and present in almost every environment. Hence, users now are offered high capabilities of computation power, battery lifetime, storage capacity, content generation tools (e.g. camera, voice recorder, etc.) and communication technologies (e.g. 3G/4G, WiFi, bluetooth, etc.).

This wide range of extra capabilities has permitted a natural evolution of the networking patterns, by which multiple devices may communicate, interact and share content. Alongside the standard infrastructural communication models (e.g. 3G/4G), devices are enabled to communicate using their built-in short range technologies. Bluetooth and IEEE 802.11 WiFi are the most common examples for establishing such an ad-hoc short range communication. Consequently, this has opened the door for a new era of mobile ad-hoc networks, shortly known as MANETs [4]. The general idea behind mobile ad-hoc networking is to offer a decentralized connection, which decouples the communicating nodes[1] from access points and base stations. That is, when two nodes are physically close enough to be in the communication range of each other,

1 Users with communication-enabled handheld devices.

they can substitute the data exchange through a centralized infrastructural entity (Figure 1.1a), with the ad-hoc short-range alternative (Figure 1.1b). This way, the two nodes can overcome possible problems associated with infrastructural networks, e.g. weak coverage, congestions, censorship, etc.



(a) Infrastructural (b) Ad-hoc

Figure 1.1: Mobile Networking Pattern: Infrastructural Vs. Ad-hoc

Moreover, MANETs are not limited to simple direct communications, which go through one hop from source to destination. A MANET communication might have a longer path, and hop through multiple intermediary nodes to reach the destination. The communication distance is measured then by the number of hops, that was required to bridge the source and distance nodes. This is why this kind of networks and all its underlying categories are referred to as *Multihop* networks [5]. Figure 1.2 depicts a simple MANET example, where nodes' intersecting communication ranges allow to establish a communication from source to destination, with a distance being equal to three.

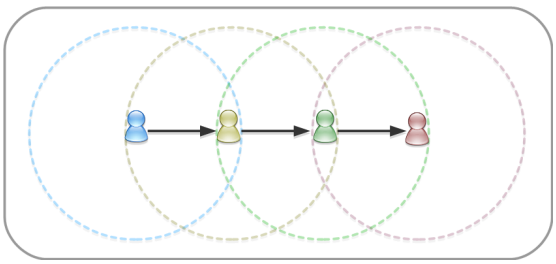


Figure 1.2: Multihop MANET

A continuous and unbroken path from source to destination, as the one shown in Figure 1.2, is not only an option for MANETs but a functioning condition. An end-to-end path, regardless the number of hops, must be available for a message to be successfully delivered. Any disconnection on the path, due to nodes' mobility and topology changes, will result in a transmission failure.

To tolerate such disconnections and resulting delays, Delay Tolerant Networks (DTNs) and Opportunistic Networks (OppNets) emerged. When the occurrence and location of disconnections are predictable, DTNs can push more tolerance into the network functionality by allowing specific intermediary nodes to carry received messages until having the opportunity to forward them again. This behavior is known as the Store-Carry-Forward (SCF) paradigm. It lays the end-to-end connectivity condition aside, and introduces a new form of ad-hoc networks based on delay tolerance and opportunistic message forwarding [6, 5]. The corresponding functionality is expressed in Figure 1.3 through timely separated snapshots. It shows how an intermediary node may deliver a message from source to destination without having to be connected with both at the same time. Mobility makes it possible to bridge nodes temporally, even if they are spatially disconnected.

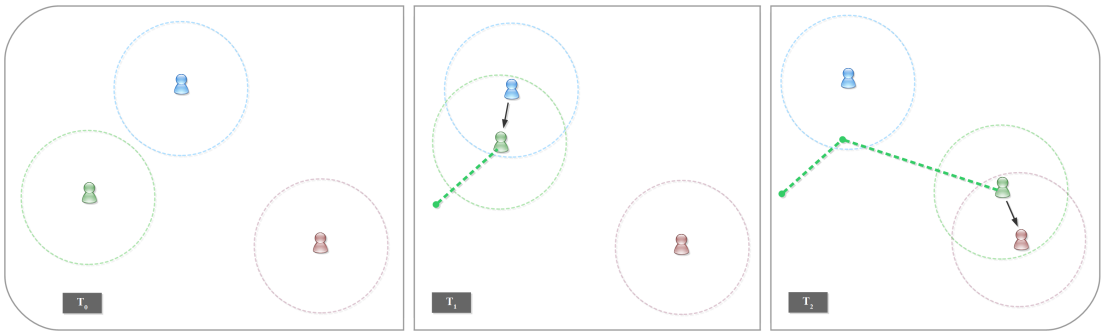


Figure 1.3: Functionality of The Store-Carry-Forward Paradigm

Otherwise, when the occurrence and location of disconnections are unpredictable, Opportunistic Networks (OppNets) allow the network to function in a fully dynamic fashion. OppNets are regarded as multihop networks with a valid Store-Carry-Forward (SCF) paradigm for any arbitrary number of intermediary nodes before reaching the destination. Consequences of the offered dynamicity and flexibility include advantages and disadvantages. On the first hand, OppNets support indirect end-to-end data delivery and possess a wider communication range, that is given by the dynamic network topology. Nevertheless, on the other hand, because forwarding opportunities depend on nodes' mobility, communication disruptions and uncertain data delivery are dominant. These characteristics make the usual use case scenarios of OppNets limited to the exchange of small messages, in relation for example to traffic notifications or disaster alarms [5].

1.1.2 *Target Applications*

Beside the networking pattern, the availability of highly capable handheld devices has also revolutionized the content generation prospects through the simplification of producing, storing and sharing audio/visual content.

Sharing multimedia User-Generated Content (UGC) can take place either on social networks like YouTube and Facebook, or using messaging services and applications like MMS, WhatsApp and Google Hangouts, which all go through infrastructural communications. Nevertheless, special scenarios or situations for multimedia dissemination do exist, where users need to lay infrastructural means aside and switch to a multihop ad-hoc alternative. Examples are when infrastructural communications are subject to undesired censorship, are not available, or apply extra costs.

1.1.2.1 *Censorship*

Infrastructural communications can be easily subject to undesired censorship by third parties, including but not limited to dictator governments. Any data exchange must go through central service providers, which can analyze the data and identify the concerned users.
Consequences of detecting sensitive data may range from a simple denial of service to prevent data delivery, to extremely pursue and detain people. Therefore, OppNets can provide an ad-hoc alternative that lays central service providers aside.
Example scenario: Scattered fleeing anti-government demonstrators, sharing sensitive content that must be kept out of censors' reach.

1.1.2.2 *Connection Absence*

Despite the massive deployment of cellular infrastructures, a network connection like 3G/4G might fail in far rural areas due to coverage problems, or in over-crowded urban areas due to network congestions.
Example scenario: Participants in a nature excursion in a rural geographical area, sharing image/video shots. The area is not covered by an infrastructural communication, and an ad-hoc connection has to be established.
Furthermore, another possibility is the collapse of infrastructural network in emergency situations due to direct damages.
Example scenario: Network providers or power supplies crash in a war zone, a terrorist attack or a natural disaster hit.
In both cases, OppNets provide an ad-hoc alternative that can still function with the aid of the mobile users moving in, to and out of the concerned area.

1.1.2.3 *Over-Costs*

Data usage is usually size- or time-limited by mobile providers or public access points, which does not fit for exchanging large multimedia content. A worse case is when a user is roaming, where extra fees apply.
Example scenario: Foreign tourists (on roaming), sharing image/video shots.

So, even if the area is covered by an infrastructural communication, OppNets provide a free access to an ad-hoc network that can be established anytime and anywhere.

1.1.2.4 *Summary*

Motivated by those example scenarios, this dissertation focuses on designing a video dissemination scheme in Opportunistic Networks within the scope of the defined research problems in Section 1.2.

1.2 RESEARCH PROBLEMS

OppNets are human-centric, that is, data forwarding opportunities tightly depend on how humans move and come in contact with each other. OppNets do not impose strict preconditions as a continuous end-to-end connectivity to ensure data delivery, assuming that the Store-Carry-Forward (SCF) paradigm will overcome the disconnections on the multihop delivery path. Consequently, challenging specificities must be taken into account when designing any communication scheme in the context of OppNets. These include unanticipated delivery delays, high loss ratios and an unstable connectivity due to the continuously changing network conditions, which affect the contact patterns among nodes, and hence the forwarding opportunities. Moreover, implementing a video dissemination scheme in OppNets also involves new design and evaluation requirements. First, OppNets would expose large-sized video data to partial losses, which can result in whole failures because of the continuity of the video medium. Second, the video viewing QoE can not be easily implemented by the means of the perceived video quality because of the presence of other important criteria, as the varying resources (e.g. bandwidth and storage) and the varying delivery conditions (e.g. delivery ratio and delay).
Therefore, the broad research problem at the core of this thesis is:

(P) *How to design and evaluate a video dissemination scheme for infrastructure-less Opportunistic Networks?*

Where the term "dissemination" in this context refers to the category of routing techniques for OppNets, which is defined by Pelusi et al. [5] as a form of "controlled flooding." On the contrary to the classical meaning, this does not necessarily involve broadcasting to multiple recipients, but a single recipient is acceptable according to this definition.
The challenging specificities of OppNets, along with the videos' large volumes, present the fragmentation[2] as a natural candidate for the solution. It allows to reduce the transmission units' volumes, to split large video content into smaller chunks and then to reach the destination through different paths and at different timestamps.

2 or partitioning, or granulation.

Furthermore, depending on the coding algorithm, multimedia files can tolerate a certain amount of loss at the expense of quality, offering to view the content at multiple quality levels. Consequently, our first specific research problem can be formulated as:

(P_1) *How to granulate a video and route the resulting chunks, to improve the viewing QoE and resist partial losses?*

($P_{1.1}$) A corresponding sub-problem is then: *How to define and measure the viewing QoE?*

The importance of the adaptivity to the changing OppNet conditions arises from the fact that in more stable systems like MANETs, the setup of the fragmentation (e.g. the size and number of the smaller chunks) can be predetermined. On the contrary for OppNets, this is not possible due to the unstable connectivity and dynamically changing network conditions, like node density and mobility. Thus, the second research problem to be addressed is:

(P_2) *How to adaptively tune the size of the data chunks, with respect to OppNet's changing conditions?*

Lastly, the usual case in OppNet communication schemes is that the destination node plays a passive and inactive role, while data is pushed by the source. According to the degree of dependency between the different data chunks, small fractions of partial losses may put the destination (receiving node) into a long or endless suspension. Hence, we formulate our last research problem as:

(P_3) *How to design a loss concealment mechanism at the destination node, to enable reacting to small amounts of loss?*

1.3 OBJECTIVES & HYPOTHESIS

Concluding from the last section, our objectives regarding a video dissemination scheme in Opportunistic Networks are:

EFFECTIVENESS manifests itself by two parameters, delivery ratio and delay, both objectively affecting the viewing experience of the video upon reception.

1. Delivery ratio: is the percentage of the delivered data, to the intended amount of data to be delivered.
2. Delivery delay: is the waiting time between the start of sending a video by the source, and when it can be viewed by the destination.

EFFICIENCY manifests itself by the overhead, which is measured differently according to the transmission model and the routing technique. In dissemination-based routing, as defined above, the overhead is given by the Equation 1.1.

$$overhead = \frac{I_{data}}{D_{data}} \tag{1.1}$$

where

D_{data} : The amount of successfully delivered data

I_{data} : The initiated amount of data, including the redundancy of the "controlled-flooding" routing

Toward these objectives, the thesis argues that applying a scalable layering video coding, which fragments videos into parts (called layers) representing overlaid levels of quality, is the first clue of the solution.

1.3.1 *Assumptions*

The research area of Opportunistic Networks is too broad, and therefore it must be narrowed by drawing constraints and assumptions on some aspects, which can not be addressed in this thesis.

- Energy resources: optimizing the energy consumption in ad-hoc network is an independent research area, and hence not considered in our contribution.
- Cooperation: participating nodes are assumed to be cooperative by nature, without a need to implement a framework for incentives or consider message drops at "malicious" nodes.
- Security: Disseminated content is assumed to be safe from dropping, eavesdropping, intrusion and forgery.
- Social network: no overlaid social network is assumed to be present beside the ad-hoc network of the participating nodes.

1.4 THESIS CONTRIBUTIONS

Conforming with the presented research problems in Section 1.2, this thesis contributes to the field of Opportunistic Networks by designing and evaluating a communication scheme for video dissemination in a fully infrastructure-less and delay-tolerant context. This scheme covers the following contributions:

(C_1) The scheme applies a scalable video coding technique, which granulates videos into scalable layers of smaller volumes, holding different levels of quality. The dependency among layers marks them with multiple unequal importance degrees. This property is exploited to prioritize the video layers and route them with unequal degrees of redundancy proportionally to their importance. As a consequence, a reasonable trade-off between the delay and quality parameters of the viewing experience is obtained. The playout delay is reduced by stressing on faster and more reliable delivery of layers with lower quality (but higher importance). Moreover, the quality parameter is not laid aside, thanks to the space for later arriving quality-enhancing layers.
In order to evaluate (and compare) the different measurement parameters on a single basis, a definition of the viewing QoE is proposed. With the aid of this definition, delivering an objectively better video viewing experience is argued, in comparison with the dissemination of non-scalable video within the same context.
Relevant publication: Klaghstan et al. [7].

(C_2) The scheme takes the fragmentation into a deeper level, by not being limited to the scalable layers (that have predetermined sizes) as the granules[3]. The network's varying conditions cause fixed-size layers to fall into one of two problems regarding the durations of data forwarding opportunities:

- either a layer is too big to be sent during the given forwarding opportunity, so a transmission failure will occur.
- or it is too small, so it is successfully sent during the given forwarding opportunity, which is still open but is not sufficiently long to send a complete second layer. Hence the opportunity is not optimally exploited.

Therefore, the scheme we propose enables each node to record a history of environmental information regarding the contacts and forwarding opportunities, and use this history to predict future opportunities and optimize the units' sizes accordingly. Consequently, by complementing the solution of (C_1) with the new proposal of (C_2), the video viewing QoE is enhanced in terms of both data delivery and delay.
Relevant publication: Klaghstan et al. [8].

(C_3) The scheme involves the destination node by implementing a loss concealment mechanism at its side, with the capability to react to an amount of loss. The implemented concealment mechanism functions in a composite manner with a twofold profit: 1) a content-based video frame loss concealment, and 2) a network-demands-based loss concealment, where demands are initiated and

3 or fragments, i.e. the individual composing units.

cast to other nodes in the network in search for the missing data parts.
This part of the solution joins the prior two ones as a second step that comes afterwards. The receiver node starts receiving data parts that were prepared by the source node using (C_1) and (C_2). Then, it applies (C_3) to recover the few and small (however critical) data parts that failed to arrive in the first step. Implementing this mechanism at the receiver side helped to enhance the video viewing QoE through a better data delivery.
Relevant publication: Klaghstan et al. [9].

1.5 THESIS ORGANIZATION

The next chapters of the thesis are organized as the following:

CHAPTER 2: highlights the theoretical background of the addressed themes in this thesis, including:

- The target domain of ad-hoc networks.
- The simulation modeling for this type of networks.
- The fundamentals of video coding and its available techniques.

CHAPTER 3: discusses works in the literature related to our research. It starts with a general overview of state-of-the-art video communication schemes in multihop networks. Then it focuses more on the works that are directly related to the identified research problems.

CHAPTER 4: presents the first contribution (C_1), which focuses on video granulation to shift the transmission units from entire videos to smaller parts. It also defines the viewing-experience evaluation metric.

CHAPTER 5: presents the second contribution (C_2), which aims to vary and tune the sizes of the video data chunks in a dynamic and adaptive way according to the OppNets changing conditions.

CHAPTER 6: presents the third contribution (C_3), which aims to induct a concealment mechanism at the destination node.

CHAPTER 7: concludes this work by giving a short summary of the addressed issues, and suggesting possible directions of future research.

APPENDIX A: overviews the essential technical material that enables delivering the scientific contribution of this thesis. It includes instructions, implementations and techniques from different addressed aspects.

* * *

Chapter 2

THEMATIC BACKGROUND

This chapter highlights the theoretical background of the topics addressed in this thesis: Section 2.1 discusses the target domain of ad-hoc networks, and Section 2.2 presents the simulation modeling for this type of networks. Lastly, Section 2.3 tackles the fundamentals of video coding and its available techniques.

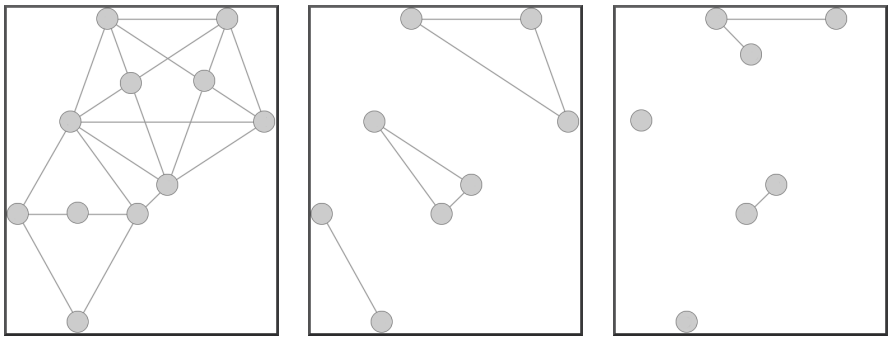
2.1 OPPORTUNISTIC NETWORKS

This section aims first to precisely define Opportunistic Networks (OppNets) against other types of multihop networks, and derive key differences among them. Then, it discusses available data delivery mechanisms in OppNets and their corresponding applications and requirements.

2.1.1 *Multihop Networks: Terminology & Taxonomy*

As presented in Chapter 1, the massive growth of mobile devices and their built-in technologies has allowed the emergence of autonomous multihop ad-hoc networks beside their infrastructural counterparts. The advancement of applications from early military scenarios to more diverse civil ones [10], introduced new adaptation requirements and forked the multihop networking paradigm into Mobile Ad-hoc Networks (MANETs) and Delay Tolerant Networks (DTNs), and the latter differentiated then between DTNs and Opportunistic Networks (OppNets) [5].
However, these three subclasses of multihop networks are not always precisely separated in the literature, and sometimes interchangeably misused because of some similarities and common characteristics. For example, all the three types support mobility; MANETs and DTNs may include sub-communities; and DTNs and OppNets rely on the Store-Carry-Forward (SCF) paradigm. Nevertheless, we argue in this thesis that the three subclasses of multihop networks have different properties in term of connectivity, data delivery and network topology. Pelusi et al. [5] trace all these differences back to a single factor: the density, which by ranging its value higher or lower

may alter between the different forms of multihop networks. Krishnamachari et al. [11] represent wireless networks as random graphs, with vertices referring to mobile nodes and edges to established connections. The density of the network, given by the density of the graph edges, is classified into three levels: dense, medium and sparse, which are visually illustrated[1] in Figure 2.1.



(a) Dense (b) Medium (c) Sparse

Figure 2.1: Density Patterns of Wireless Networks

This classification is also compatible with the network partitioning patterns proposed by Spyropoulos et al. [13]. These latter claim that according to the mobility and transmission ranges, a wireless network may fall into one of the following categories:

1. Well-connected: The network is very dense, and shape an almost complete graph. A continuous path between any two nodes is most likely to exist, which fulfills the end-to-end connectivity requirement of MANETs.

2. Connectivity islands: The network is composed of diverse connected partitions, with no spatial direct interconnections between them. The possibility for a node to move from one partition to another interconnects them temporally, and shapes the core of DTNs.

3. Sparse: The network is composed of individual nodes, who meet only opportunistically. Therefore, at any instant of time, there is only a small number of active connections. OppNets fall in this category.

2.1.1.1 *Mobile Ad-hoc Networks (MANETs)*

MANETs are defined as autonomous self-organizing networks, composed by mobile nodes with short range ad-hoc communications [10]. Data delivery in MANETs can go through a number of hops from source to destination, on a continuous path that has to be planned prior to launching the delivery process, and is able to end-to-end connect the source and the destination together. Any member node in the network

1 Illustrations generated on GraphTea [12]

may be employed as a relay for the good of messages delivery. An example of this is illustrated in Figure 2.2, with a source, a destination and two relay nodes, all within the corresponding communication ranges.

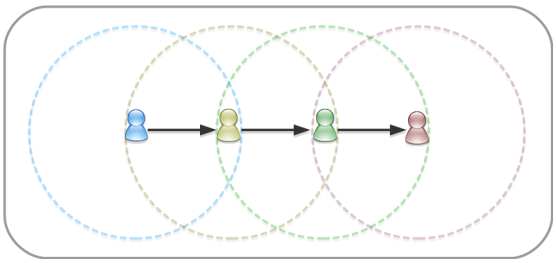


Figure 2.2: Multihop Data Delivery in MANET

However, nodes' mobility is unlimited, so that nodes can freely move in the environment and consequently change connection pairs in the communication range. MANETs regard such network topology changes and resulting disconnections as disruptions that eliminate the already planned end-to-end paths, and necessitate planning new ones [14].

TOPOLOGY Although MANET's topology may rapidly and frequently change, new distributions and nodes have to be recognized in order to be able to establish new end-to-end paths [15]. Since MANETs assume a regular presence of end-to-end paths and an immediate update of these paths upon topology changes, the network in general has to be dense and fully connected [4], as shown in Figure 2.1a.

ROUTING The necessary presence of end-to-end paths in MANETs affects the design of routing protocols. According to the way paths are established, MANET routing protocols can be classified into proactive and reactive [4, 16]. Proactive (table-driven) protocols provide each node with an updated routing information to other nodes in the network. Reactive (on-demand) protocols, on the other hand, establish the path only when a transmission has to take place. However, the common characteristic among both types of routing protocols is the TCP/IP-like design [17], which fails in the case of a sudden path break.

2.1.1.2 *Delay Tolerant Networks (DTNs)*

Delay Tolerant Networks (DTNs) differ from MANETs in their reaction behavior toward connectivity interruptions. DTNs do not assume the presence of an end-to-end connectivity, and do not regard disconnections due to mobility as disruptions. On the contrary, DTNs take advantage of users' mobility to widen the topology of the network and reach possibly far located nodes. DTNs were introduced in 2003 by Fall [6] to address applications of the so-called "challenged networks," which violate the traditional assumptions of the TCP/IP model, like stable connectivity, reliability and

constrained delays. In a DTN, the destination is not supposed to be reachable by the source on a preplanned end-to-end path. This is overcome by applying the Store-Carry-Forward (SCF) paradigm, which yields that any node receiving a message is able to store and carry it, until an opportunity of forwarding it to another node exists. This functionality is illustrated in Figure 2.3, which depicts a source, a destination and an intermediary node that comes in contact with them at different time instants.

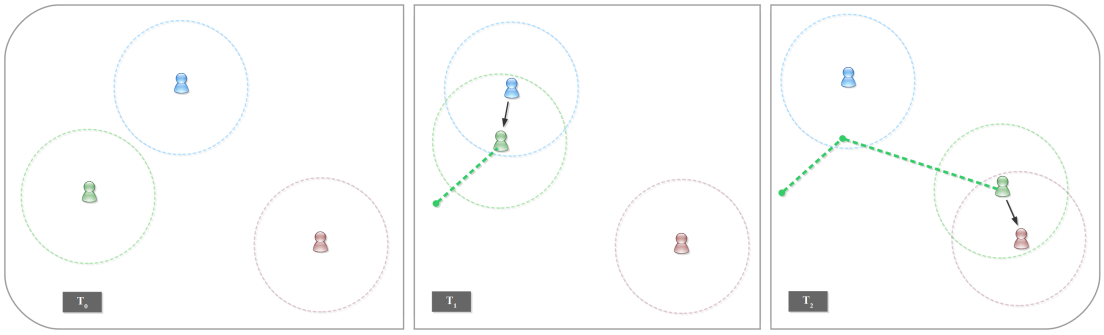


Figure 2.3: Functionality of Store-Carry-Forward

TOPOLOGY DTN's overall topology is split into subnetworks, that are called "islands" as in [13], "regions" as in [6], or "communities" when they refer to social relationships. Each of these subcomponents has an ongoing intra-connectivity within [5, 6, 13], but inter-connections between them can take place only indirectly through the SCF paradigm. DTNs assume that each island can predict and localize disconnections [15]. This property affects the design of routing and data delivery protocols.

ROUTING The purpose of DTNs is to provide interoperability between the diverse disconnected subnetworks, regardless of the intra-connection within each subnetwork [6]. Since the network is assumed to be able to predict the locations of possible disconnections, it setups specific nodes as "gateways" at those points, in order to connect the separated regions through their mobility [15, 6, 13]. Once the assumption of disconnections predictability is no longer valid, the domain of the target multihop ad-hoc network moves to OppNets.

2.1.1.3 *Opportunistic Networks (OppNets)*

Opportunistic Networks (OppNets) share their basic characteristics with DTNs, so that earlier in the literature both terms were often used to refer to the same concept of networks [14, 18]. However, Pelusi et al. [5] introduce OppNets as being derived from DTNs, but representing a more flexible form of it. Similarly to DTNs, OppNets do not assume an end-to-end connectivity as a precondition for data delivery. They con-

sider users' mobility as an advantage, and tolerate resulted disconnections via SCF paradigm. Nevertheless, unlike DTN's subcomponents and their predicable disconnections, OppNets assume zero-knowledge about the network topology, which makes each relay node responsible for deciding on the next hop for routing based on local information only (e.g. previous and currently available contacts) [5, 19]. While the objective in DTN is to interconnect heterogeneous connectivity islands, nodes in OppNet are standalone. Consequently, OppNets can be defined based on individual-nodes, on the contrary to the network- and subnetwork-based definition of DTNs [20].

TOPOLOGY OppNet's architecture is based on the interoperability between standalone nodes, each relying only on local information with regard to previous and currently available contacts. Hence, according to Figure 2.1, OppNets belong to the third category of sparse networks [21, 22], where no end-to-end paths exist, and the individually moving nodes are the building elements. If a snapshot of the network is taken at any instant of time, it will show scattered nodes with only a few number of active connections.

ROUTING Contrarily to DTN, relaying messages in OppNet is not limited to specific gateway nodes. Because path breaks and disconnections may unpredictably take place anywhere in the network, any node in OppNet can play the role of a relay [5, 18]. However, once a message leaves a DTN connectivity island on a gateway node, the routing between intermediary nodes will be identical to how it works in OppNet. Therefore, both DTNs and OppNets share the same literature on routing strategies (as will be presented next).

2.1.1.4 *Summary & Discussion*

The outlined differences and characteristics of the multihop networks subclasses are summarized in Table 2.1.

	MANET	DTN	OppNet
Topology	Dense	Connectivity islands or partitions	Sparse with standalone nodes
Connectivity	end-to-end	Store-Carry-Forward	Store-Carry-Forward
Routing	Full knowledge	Basic graph-based knowledge	Zero previous-knowledge, Opportunistic contacts
Relay	Any node on a continuous path	Definite nodes on definite locations	Any node

Table 2.1: Multihop Networks

This thesis is based on sparse OppNets with standalone nodes, that meet only oppor-

tunistically and deliver data on top of the SCF paradigm. This allows to study the delivery application under extreme conditions, which are not unlikely to happen.

2.1.2 *Data Delivery*

A large amount of works in the literature regarding data delivery and routing protocols for OppNets is available. Furthermore, a large part of this literature is shared with DTN because both types of networks function under the same philosophy of SCF and indirect end-to-end connectivity. Accordingly, a number of surveys have been published to summarize and categorize these works [13, 5, 18, 23, 24, 25, 26, 27]. Classifications in [5, 18, 24, 27] base on two criteria: replication and knowledge, to split data delivery approaches in OppNets into dissemination (flooding)-based and forward-based.

2.1.2.1 *Dissemination-based Routing*

This type of routing is also called replication-based [13, 23] or flooding-based [18, 24] routing, and it aims to spread messages to nodes in the network using replication [5]. Redundancy is meant to compensate the absence of advanced knowledge about the network. Two extreme cases of this approach are implemented by tunning the number of message copies to be distributed in the network, and the number of hops, over which a message is allowed to move. The two implementations are the *direct delivery* and *epidemic* routing protocols.

- Direct delivery: as a lower bound, this protocol involves a singly copy of source messages, that are forced to be kept at the source node until meeting the destination, to be directly delivered in a single hop. This ensures the best overhead and resources utilization, but the worst delivery probability and delay [28].
- Epidemic: implements the upper bound by an unlimited flooding [29]. In epidemic routing, a node carrying a message, replicates it to any other encountered node limitlessly, unless that node already has a copy of the message (avoid cycles) or the Time-to-Live (TTL) parameter of the message expired. Consequently, the message will spread as a disease, whose infection is a new replica of the message. This obviously enables the best possible delivery ratio and delay, but it is very resource-hungry and causes maximum overhead.

All other routing protocols lay in between these two extremes, trying to control flooding in order to trade-off delay and delivery ratio on the first hand, with resources consumption (bandwidth, storage and energy) on the other hand. To achieve that, two strategies are possible: replication control and estimation functions [13, 18].

REPLICATION CONTROL As with epidemic routing, messages are copied when nodes come in contact with each other. The difference is that the number of replicas

of a given message that may exist at a given time is bounded my a maximum value. Spyropoulos et al. [30] propose the Spray-and-Wait (SnW) protocol, which consists (as the name suggests) of two phases: spray and wait. During the first phase, k replicas of a message are "sprayed" to k nodes in the network. Then, these nodes "wait" until they come in contact with the destination to deliver the message. The spray phase has two variations: Vanilla and Binary. Vanilla spray replicates the message to the first k met nodes, with one copy to each. Whereas Binary spray suggests to hand over half of the copies upon each contact. Each node repeats the same until it has only one copy left, where it enters in the waiting phase.
Another possibility to control replication is to bound the number of hops, that a message is allowed to go through. This is like the already mentioned direct delivery routing protocol.

ESTIMATION FUNCTION In this strategy, the replication decision is not limited by a maximum number of replicas, but by an estimation function, which determines whether to exploit the contact opportunity or not. This function is evaluated based on narrow simple knowledge regarding the nature of replicated messages or the fitness of encountered nodes. Ramanathan et al. [31] and Wang et al. [32] propose prioritized epidemic routing protocols, which prioritize messages based on local information like messages' TTL or inter-node exchanged simple information to reflect messages' delivery costs. On the other hand, the PRoPHET routing protocol [33] states that when two nodes meet, they exchange summary vectors of encounters history, which indicates to each other their delivery predictability to a given destination.

2.1.2.2 *Forward-based Routing*

The forward-based routing protocols need more information about the network topology. Furthermore, for a given message, no replicas are generated in the network but the message itself is handed over along the possibly best determined path. When nodes meet, a forwarding decision has to be taken, which is conceptually similar to the estimation function of dissemination-based routing. The difference is that a forwarding decision is not concerned with whether to replicate a message or not, but with entirely handing the message over. Therefore, the decision is more sensitive, and must be based on available contextual knowledge. The simplest form of such knowledge is based on location, where forwarding takes place to a node only if is closer to the destination in a given coordination system [27]. The coordination system here can have a physical perspective, as for instance if GPS is enabled, or a conceptual perspective as proposed by Leguay et al. [34], where the distance of two nodes represent their likelihood to meet.
A higher level of knowledge encloses more specific contextual and environmental

parameters, as in the context-aware routing protocol [35], which enables the nodes to evaluate and exchange delivery probabilities toward possible and known destinations in the network. Consequently, the current carrier of a message may decide for the best next relay based on contextual information like mobility, connectivity changes and energy level.
Forward-based routing achieves a better resources consumption, because of the limitation to a single copy of the routed messages. However, this is accompanied by increased delays and computational costs [5].

2.1.2.3 *Discussion*

Dissemination based data delivery is more common in OppNets, not only because of the direct disadvantages of forward-based routing, like delay and computational complexity, but because of the unaffordable amount of necessary contextual knowledge for forward-based strategies. This family of routing protocols was originally adopted from more stable networks, e.g. wired networks or MANETs. Therefore, they are proven to perform poorly on DTNs and OppNets [27, 5, 18].

* * *

2.2 SIMULATION MODELING

This section introduces the need for simulation modeling as an evaluation method. A number of state-of-the-art available simulators are reviewed, with a deeper look on the Opportunistic Network Environment (ONE) simulator, which was used for all the experiments in this work. Lastly, mobility models, which are an important part for the simulation modeling, are discussed.

2.2.1 *Overview*

The performance evaluation of a proposed solution in the domain of wired or wireless network can be done by three methods of evaluation [36] :

1. Complexity analysis
2. Experiments based on real-world testbeds
3. Simulations

As Lindeberg et al. [36] argue, although performing a mathematical complexity analysis to reflect the system's performance might be the cheapest variation (in terms of applicability and usability), its complexity dramatically grows with the complexity of

the system itself. Therefore, this evaluation method is incapable of modeling whole complex systems with a big number of interoperating components and parameters. Real-world experiments, on the second hand, deliver reliable and actual results but they are expensive and difficult to deploy and run. Consequently, computer-based simulations become the most used evaluation method, as they are utilized in 75-86% of the evaluation experiments, according to analysis surveys [37, 36].
A simulator is a software, which enables virtually implementing and evaluating a network on a computer [38]. It has the advantage of making it easy to rerun experiments at large scales, under different conditions or setup parameters. Furthermore, within the constrained environment of a simulator, the comparability and repeatability of experiments are better offered [22].

2.2.1.1 *State-of-the-Art Simulators*

There are basically four simulators, that are usually used for DTN and OppNet applications [39, 40] :

- NS-2 [41] (and NS-3 [42]) : A general network simulator, and the most used simulator for MANETs in the literature [36]. However, it has only a superficial applicability for DTNs and OppNets.
- OMNeT++ [43] : Similarly to NS-2, it is a general network simulator, based on a component and modular architecture. It also lacks good support for DTN.
- DTNSim [15] : It is developed exclusively for DTNs. It simulates message routing using SCF, but needs input data of connectivity traces to determine availability and unavailability of links between nodes.
- ONE [1] : A recent simulator for DTNs and OppNets. It supports the simulation of mobility and routing, in addition to events visualization. Since its launch in 2009, it has obtained a significant interest and a wide usage in the literature [21].

2.2.2 *The ONE Simulator*

2.2.2.1 *Architecture*

Opportunistic Network Environment (ONE) is an open-source Java-based simulator, designed to implement and evaluate applications in DTN and OppNet [1, 44]. ONE is a discrete event simulator, that is, simulation's functional modules are updated at discontinuous simulation steps. As shown in Figure 2.4, ONE has four interoperating functional modules:

1. Movement models

2. Routing
3. Output visualization/reports
4. Event generators

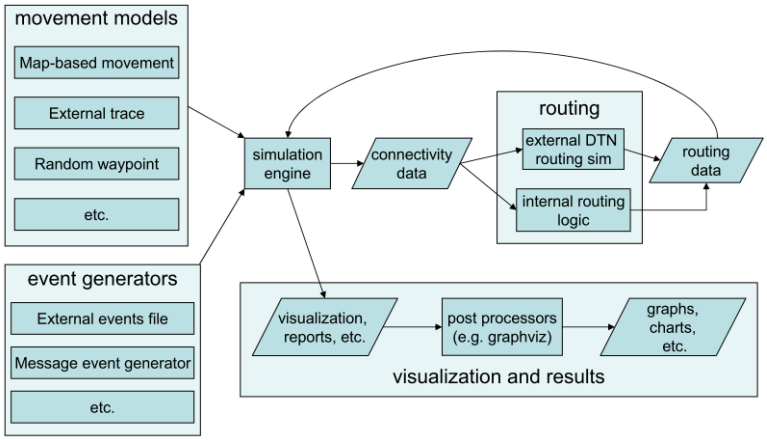


Figure 2.4: ONE Simulator's Functional Modules [1]

The movement model is responsible for controlling nodes' positions in the simulated environment, and hence determining whether they are in the communication range of each other to exchange data. This information is fed into the routing module to know how to hand over messages, that were created in the event generator module. Lastly, ongoing actions can be followed graphically on the simulator's GUI (as in Figure 2.5), or they can be written onto output textual reports for further processing.

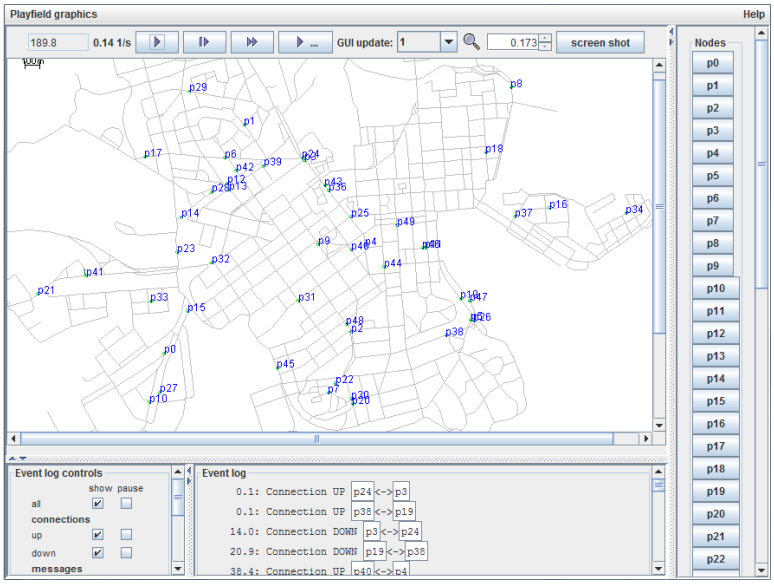


Figure 2.5: ONE Simulator's GUI

2.2.2.2 *Advantages of ONE*

ONE decouples the simulations from the low-level implementation details of the wireless communication, which results in a simpler communication model between nodes that is based on only being in the reach of each other. This, according to Keränen et al. [1], "allows focusing on the evaluation of the DTN protocols, instead of fully modeling the lower layers." Furthermore, the focus on the usage scenarios of DTNs and OppNets, frees the simulator from the complications associated with general networks support. Lastly, the open-source and modular architecture of the simulator supports extensibility and customizability of its functions to a large extent.

2.2.3 *Mobility Models*

The functionality of the movement module is based on the mobility models, which defines how the nodes move in the simulation environment. According to their generation, there are two types of mobility models: synthetic and based on real-world traces [45].

2.2.3.1 *Synthetic Mobility Models*

These are theoretical models, and as in [40, 1] they are further classified into:

1. **Random:** Mobility is generated using mathematical random functions. Examples are the Random Walk (RW) and Random Way-Point (RWP) models [46], where initial starting points, drawn destinations and movement speed are all randomly assigned. Despite its unreality and limited performance, RWP is the most used model[2] in the simulations due to its simplicity [36]. Consequently, it has become a common comparison reference in the literature.

2. **Real Mobility-Driven:** To reduce randomness and shift the model to the real-world case, mobility models can be based on real maps or real human behaviors. Map-based models derive movement paths from data on real maps. The most common examples are the Map-based Movement (MBM) and Shortest Path Map-Based Movement (SPMBM) [1]. Human behavior models are a further step, which include daily human actions, like the Working Day Movement (WDM) model [47].
   MBM is similar in its core to RWP, except that the movement must follow the paths on the defined map. SPMBM is a bit more realistic, in the sense that nodes select a point on the map either randomly or from a list of interesting points, then take the shortest path on the map to that point. On the other hand, WDM

2 This is to be shown later in Table 3.2

tries to better mimic reality by modeling human activities like sleeping, working and going out, each with a different corresponding location.

2.2.3.2 *Real-World Traces-Based Mobility Models*

Mobility models, that are based on this kind of traces, do not involve any mathematical estimation, but instead they use real world measurements. Experiments are performed to track real users and collect data, that reflect different aspects of nodes' mobility and contact [40].
Although the trace-based mobility models obviously yield the most realistic setup, their main drawback is that they are collected from very specific scenarios with limited populations, and therefore they can not be reliably generalized. Furthermore, real world traces are characterized by low temporal or spatial resolutions [44, 40]. That is, frequency of scans for other devices is kept low to save battery, which results in inaccurate contacts measurements as well as missing contacts. Otherwise, locations are detected by widely separated fixed Access Points (APs).

2.2.4 *Discussion*

Against the less appropriate alternatives, simulation modeling is chosen to evaluate the proposals of this work. Furthermore, the ONE simulator is used, most importantly because of its open-source structure, which allows a flexible adaptation of its functionality in order to implement the proposed solutions of the thesis. Adaptation examples include:

- Assigning variable redundancy parameters
- Packetize small messages in one big message (and depacketize later)
- Request/response of additional messages at runtime

Regarding the mobility model, two models were utilized in the simulations. RWP, to ensure a comparison reference, and to obtain a more realistic results, simulations also included SPMBM.

* * *

2.3 VIDEO CODING

This section discusses video coding from the perspectives of its purposes, types and fundamentals. Then, Scalable Video Coding (SVC), which was chosen to fulfill the need for a multistream coding, is addressed in details. Lastly, an overview of video quality measurements is given.

2.3.1 *Overview*

2.3.1.1 *Video Coding Evolution and Fundamentals*

Video coding is a critical procedure to be performed at the video source for two purposes: first, to reduce the stream bitrate for both storage and transmission purposes, and second, to enhance the reliability against possible losses or errors. The basic idea is to exploit the redundancy within a frame or between consecutive frames, in order to compress the data through redundancy reduction, and enhance the stream with error detection and recovery capabilities.
The evolution of standards for video coding was mainly defined by two organizations: the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), and the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) and its underlying Moving Pictures Experts Group (MPEG). Both organizations developed many coding standards, one of the last of which is the H.264/AVC (also called MPEG-4 Part-10), which was jointly developed to achieve high video quality at low video bitrates [48]. To control redundancy reduction and data compression, Advanced Video Coding (AVC), as an MPEG and H.26X standard, divides the sequence of frames of a video stream into a set of Group-of-Pictures (GoP). Hence, the redundancy between consecutive frames is studied within the same GoP. Frames in a GoP can be in one of three types according to their coding/decoding dependability:

- **I-frame:** Intra-frames are intra-coded independently from other frames, so that data compression only considers spatial information withing the same frame.
- **P-frame:** Predictive-frames are coded based on the previous reference frame, which is the last appearing I- or P-frame.
- **B-frame:** Bidirectional-frames are also dependently coded, but based on both previous and following reference frames, which are the last appeared and next appearing I- or P-frames respectively.

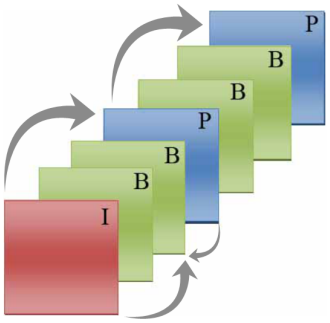


Figure 2.6: I-, B- and P-frames in a GoP

Because of this dependency, the order of frames within a GoP affects the coding complexity and the output stream size and quality, since only the difference from the reference frame has to be coded. For instance, less I-frames would result in a smaller size, but also in a lower quality and a more complex coding process. An example of a GoP with dependencies between frames is depicted in Figure 2.6.

2.3.1.2 *Multistream Video Coding*

Contrarily to ordinary coding standards, whose output is a compressed single stream, in multistream coding a video is encoded into several scalable substreams. These substreams can then build on top of each other, to deliver all together the video's overall quality. The removal of one or more substreams would result in only degrading the video quality, instead of a whole failures as with single stream coding. Consequently, parts of the main stream can be easily removed, in order to adapt to changing conditions like heterogeneous recipient devices or varying bandwidth availability. This property is referred to as "encode once, decode many." That is, when facing new demands due to conditions changes, the same encoded video can be differently decoded as many times as needed, to cope with the corresponding demands. Re-encoding or the initiation of different representations of the same content are not necessary. Moreover, the choice of scalable multistream coding has also the advantage of fulfilling the requirement of granulation. But unlike generic granulation techniques, which split the data into meaningless parts and condition reconstructing the original data by receiving them all, scalable multistream coding offers multi-level reconstruction. The overlaid parts can reconstruct the video body on levels with ascending quality, depending on the amount of data received.
For the purpose of scalable multistream coding, two approaches exist: Layered Coding and Multiple Description Coding (MDC).

LAYERED CODING Using this approach, the video content is coded into a number of stacked layers, each of which refers to a quality level. These layers reflect a back dependency property, which tightly links the decodability of a given layer with the reception of its precedent layers. Hence, the layers reflect different importance degrees, where the first layer (called "Base Layer") will possess the highest importance among other layers. The Scalable Video Coding (SVC), an extension of the H.264/AVC, is a state-of-the-art implemented example in this domain, and will be presented in details next.

MULTIPLE DESCRIPTION CODING Contrary to layered coding, MDC splits a video stream into equally important substreams, called descriptions [49]. A base quality is obtained by decoding any description, then, the more descriptions are successfully received, the higher the video quality is.

A number of comparisons are performed between the two coding approaches [50, 51, 52, 53]. Although conclusions outline close performance regarding many metrics, like delivered quality and output source size [50], other metrics reflect remarkable differences. With regard to delivery ratio and perceived video quality, MDC has an advantage unless the layered coding powerfully protects the base layer [51, 52], in addition to a higher coding complexity of MDC [51]. Furthermore, SVC (as a layered coding technique) introduces additional advantages by the means of a further partitioning of the scalability layers into content-dependent very small units. This property will be fully utilized (as will be presented later) for adaptive routing and error concealment solutions.

2.3.2 *Scalable Video Coding*

2.3.2.1 *History*

Scalable Video Coding (SVC) is one of the recent stable video coding standards, proposed as an extension to H.264/AVC [54]. It was developed by the Joint Video Team, which embraces both ITU-T's Video Coding Expert Group (VCEP) and ISO/IEC's MPEG [55]. An official reference implementation was also provided, under the name of Joint Scalable Video Model (JSVM) software [2].
A more recent successor to H.264/AVC is the High Efficiency Video Coding (HEVC) standard, which was completed in early 2013 [56]. A set of extensions, including the Scalable HEVC (SHVC), were standardized later in 2013 [57]. However, the SHVC Test Model (SHM) reference software was until very recently under development. The first conformance test draft appeared in late 2014 with the last stable version of SHM [58]. This very recent emergence of the standard prevented an accordant consideration in this work.

2.3.2.2 *Fundamentals and Objectives*

The objective of SVC is to compress and encode a raw video file with a single bit-stream into another compressed form containing many sub bit-streams. Thus, an SVC-encoded video consists of one Base Layer (BL), L_0, and one or more Enhancement Layers (ELs), $L_1, ..., L_n$. The successful reception and decoding of the BL provides a low-quality playable version of the video. Then, further ELs would enhance the quality in one of the scalability features. SVC supports three features (or dimensions) of scalability:

- **Temporal Scalability:** A temporally scalable video can be viewed on different frame rates. Hence, a higher frame per second (fps) value is obtained with every further received temporal EL. The number of temporal layers for a given video is tightly dependent on the defined GoP size. BL delivers only one frame per GoP,

and each further temporal EL doubles the frame rate. Hence, the total number of temporal layers is equal to log(GoP).
For example, if GoP=4 frames, there will be three temporal layers: $L_{t,0}$ (with 1 frame per GoP), $L_{t,1}$ (2 frames per GoP) and $L_{t,2}$ (4 frames per GoP).
Consequently, any SVC-encoded video with GoP>1 is temporally scalable by default, with no extra encoding complexity.

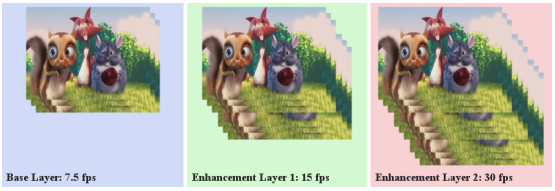


Figure 2.7: Example of Temporal Scalability

- **Spatial Scalability:** A spatially scalable video can be viewed on different spatial resolution. Spatial ELs increase the video resolution, most commonly in a 2:1 ratio between neighbor layers [59], although SVC can handle other arbitrary power-of-2 ratios [2]. That is, the spatial resolution will double with further ELs. Hence, the total number of spatial layers is determined with respect to two conditions: first, the already mentioned power-of-2 increment ratio, and second, that the resolution can not go under 96×80 [2].

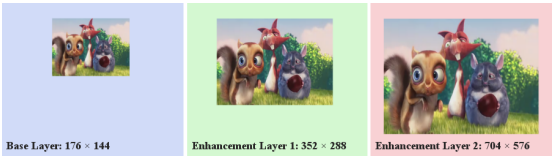


Figure 2.8: Example of Spatial Scalability

- **Quality Scalability:** This is also called fidelity scalability. It refers to the value of the Signal-to-Noise Ratio (SNR) at given temporal and spatial resolutions. A substream with a lower SNR value supports viewing a video at the given temporal and spatial resolutions but with less sharp quality, as illustrated in Figure 2.9. SVC can support up to 15 refinement layers, however this is costly with regard to coding efficiency [60].

Figure 2.9: Example of SNR Scalability

The total number of layers, n, is determined by the defined scalability levels in each of the scalability dimensions. Any layer $L_{i>0}$ is dependent on the precedent layers $L_0 \rightarrow L_{i-1}$. Consequently, layers $L_0 \rightarrow L_n$ reflect an ascending importance degree.

EXAMPLE Figure 2.10 depicts the layers' structure of an SVC-coded video, holding:

- Three frame rates: 7.5, 15 and 30 fps
- Two spatial resolutions: Common Intermediate Format (CIF) 352×288 and Quarter CIF (QCIF) 176×144
- Two SNR quality levels

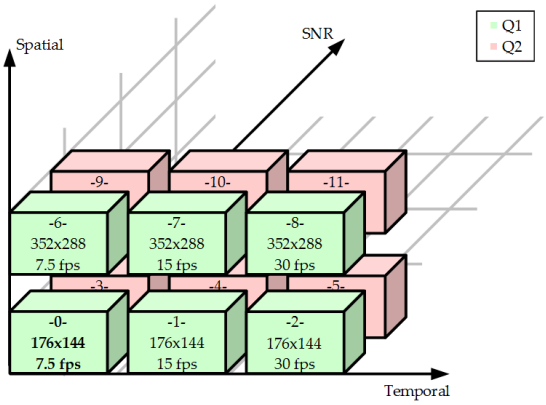


Figure 2.10: Example of SVC Layers

OPERATING POINTS SVC scalability levels allow to transform the quality for one step in one direction of the scalability dimensions. But beside this forward step in some direction, the transformation might cause backward steps in other directions. For example, in Figure 2.10, the transformation from EL-5 to EL-6 moves the quality forward in the spatial direction, but also backward in both temporal and SNR directions. To overcome this mismatch, the concept of Operating Points (OPs) is formulated to define virtual quality levels, which contain subsets of the scalability levels, and ensure an overall forward transformation in each enhancement step [60].
The number and the distribution of the OPs have many possibilities. Let us assume m, n and l are the number of the scalability levels of the temporal, spatial and SNR dimensions respectively. If $m = n = l = 1$, then the number of scalability layers: $N = 1$, and the number of the OPs: $N_{op} = 1$. Otherwise, $N > 1$ and the number of the OPs in the resulting space will be in the range:

$$2 \leqslant N_{op} \leqslant 1 + d_{Manhattan}(L_0, L_N) \tag{2.1}$$

Where $d_{Manhattan}(L_0, L_N)$ is the Manhattan distance between the first and the last layers, and is calculated with the Equation 2.2.

$$\begin{aligned} d_{Manhattan}(L_0, L_N) &= d_{Manhattan}((1,1,1),(m,n,l)) \\ &= (m-1)+(n-1)+(l-1) \end{aligned} \tag{2.2}$$

For example in Figure 2.10, $m = 3, n = 2, l = 2$, then the maximum possible number of OPs is equal to $1 + (3-1) + (2-1) + (2-1) = 5$.
Moreover, the distribution of the OPs can also vary, as different paths can be drawn from L_0 to L_N in the taxicab geometry [61]. For example, derived from Figure 2.10, two (out of many other) possible sets of OPs are depicted in Figure 2.11.

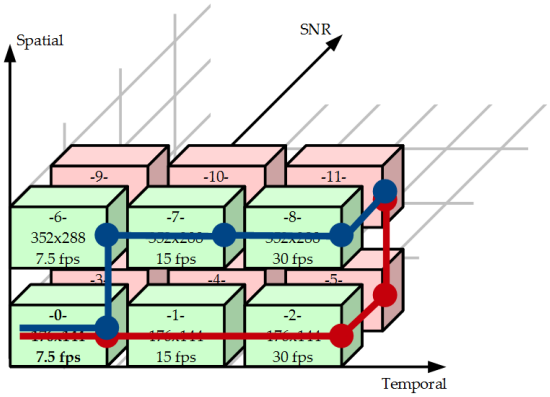


Figure 2.11: Operating Points

SET 1: depicted in red in Figure 2.11. It consists of four levels or OPs:

- Point 1: at BL
- Point 2: moves two steps in the temporal direction. Hence, it includes two enhancement layers (EL-1 and EL-2)
- Point 3: moves one step in the SNR direction, without temporal setbacks. Hence, it includes three enhancement layers (EL-3 to EL-5)
- Point 4: moves one step in the spatial direction, without temporal or SNR setbacks. Hence, it includes six enhancement layers (EL-6 to EL-11)

SET 2: depicted in blue in Figure 2.11. It consists of five levels or OPs:

- Point 1: at BL
- Point 2: moves one step in the spatial direction. Hence, it includes one enhancement layer (EL-6)
- Point 3: moves one step in the temporal direction. Hence, it includes two enhancement layers (EL-1 and EL-7)

Point 4: moves one step in the temporal direction. Hence, it includes two enhancement layers (EL-2 and EL-8)

Point 5: moves one step in the SNR direction, without temporal or spatial setbacks. Hence, it includes six enhancement layers (EL-3 to EL-5 and EL-9 to EL-11)

The choice of a specific path among other variations for a given number of OPs is critical when the sizes of the OPs are not homogeneous, i.e. there is one or more OPs on the path, whose size is extremely bigger than the others, which makes the corresponding path not preferable. Otherwise, the issue is subject to trade-offing the order of the scalability dimensions.

2.3.2.3 *Internal Structure*

The internal structure of SVC is based on a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL) [54, 55]. The coded content of SVC is represented by the VCL, which is formatted then by the NAL in a network- and storage-friendly way, by organizing the data into NAL Units (NALUs) and pushing additional information to their headers. A NALU header serves to easily extract and customize VCL content for the different possible requirements, since it contains information specifying to which scalable level the unit belongs. This is done using the following fields of the header: `temporal_id`, `dependency_id` and `quality_id`, which refer to the corresponding temporal, spatial and SNR layers respectively. SVC NALU header is four bytes long, one for the old AVC standard, and three additional bytes for SVC [60].
Partitioning the SVC content into NALUs is done in a content-wise way. That is, each unit represents one frame belonging to one of the scalability layers. Consequently, the units are different in size, depending on the content they carry [62]. NALUs of one frame among all layers are packed into a so-called Access Unit (AU).

EXAMPLE With regard to the example in Figure 2.10, the distribution of NALUs among all scalability layers, in a GoP=4 frames, would look like as in Figure 2.12.

1. The video at its BL has one frame per GoP (the lowest temporal scalability), and hence one NALU. The first and second ELs enhance the temporal resolution by adding one then two more NALUs (frames) per GoP.
2. NALUs from EL-3 to EL-5 enhance the SNR quality for the corresponding NALUs from the first step (BL, EL-1 and EL-2).
3. NALUs from EL-6 to EL-8 deliver a higher spatial resolution for the corresponding NALUs from the first step (BL, EL-1 and EL-2).
4. NALUs from EL-9 to EL-11 enhance the SNR quality for the corresponding NALUs from the last step (EL-6, EL-7 and EL-8).

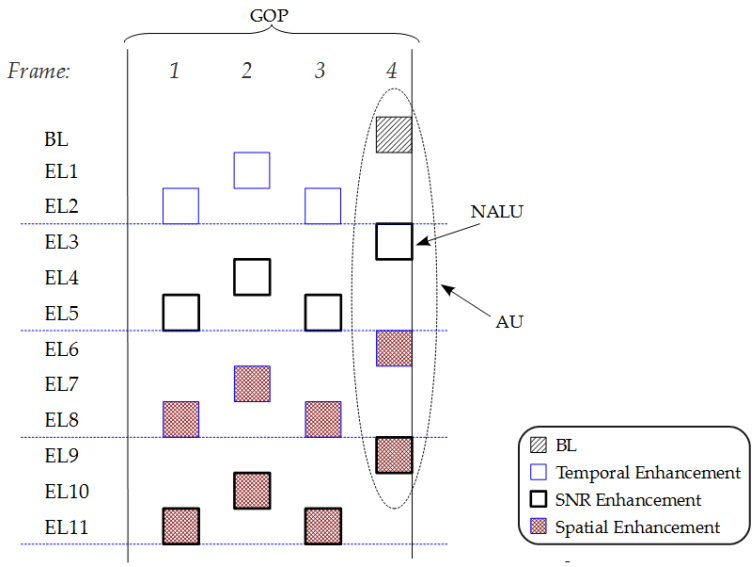


Figure 2.12: NALUs in a GoP

This architecture helps to count the number of NALUs per GoP, and hence for the whole video.

2.3.3 *Video Quality Measurement*

In network applications, the term Quality-of-Service (QoS) covers many technical performance metrics as packet-loss (the ratio of undelivered messages), overhead (the cost of delivering a message with regard to its redundancy) and delay (waiting time between initiation of message and delivery). In multimedia applications, on the other hand, Quality-of-Experience (QoE) extends QoS to measure the satisfaction of end users' experience with the perceived content quality [63, 64].
This kind of video quality measurement can be classified based on its methodology as either subjective or objective [65]. Subjective metrics reflect the quality of video perception by human viewers. They can only be measured by tests that require the participation of real human testers, which makes it expensive and time consuming [64]. On the other hand, objective metrics utilize technical parameters and mathematical models in an attempt to predict the human perception.
A commonly used subjective method is the Mean Opinion Score (MOS), which is calculated as the average subjective judgment, out of values assigned by a set of testers. MOS comes out on a scale from 1 (bad) to 5 (excellent) [66]. Objective alternatives include many methods like for example the Video Quality Metric (VQM), which measures the perceptual video deteriorations [67], the Moving Pictures Quality Metric (MPQM), which manages to ignore error pixels if they are not perceived by the human visual system [68], the Structural Similarity Index (SSIM), which tries to cap-

ture the structural distortions for a better correlation with the subjective techniques [69], and the well known Peak Signal-to-Noise Ratio (PSNR) function, which is calculated for each frame of the video with regard to the Mean Square Error (MSE) [70] :

$$MSE_{frame(i)} = \frac{1}{W.H} \sum^{W} \sum^{H} [Y_r(x,y) - Y_p(x,y)]^2 \tag{2.3}$$

$$PSNR_{frame(i)} = 10 \times \log_{10} \frac{I^2}{MSE} \tag{2.4}$$

$$PSNR_{total} = \frac{1}{N} \sum^{N} PSNR \tag{2.5}$$

where

W, H	: Frame's width and height
Y_r	: Luminance value of the reference frame in pixel (x, y)
Y_p	: Luminance value of the processed frame in pixel (x, y)
I	: Maximum luminance value
N	: Number of frames in the video

Compared to the other objective metrics[3] and the expensive subjective ones, PSNR has achieved a high popularity due to its fast and simple computation and understanding [70]. Consequently, PSNR has become the most commonly used quality measurement in the literature [36]. Moreover, a conversion between PSNR and MOS is proposed by Klaue et al. [65] based on the range of the PSNR value. The conversion values are shown in Table 2.2.

PSNR	MOS
>37	5: Excellent
[31, 37[	4: Good
[25, 31[	3: Fair
[20, 25[	2: Poor
< 20	1: Bad

Table 2.2: PSNR to MOS

For multistream videos, PSNR is calculated for each of the substreams separately[4]. The overall PSNR is obtained then by considering the video with all its substreams present.

3 Available objective metrics in the literature are much more than the above mentioned [71].
4 More details in Appendix A.2.

2.3.4 *Discussion*

This work is based on the choice of scalable multistream coding, which offers two advantages:

FIRST, the built-in fragmentation into smaller parts, which eases the delivery of larger data.

SECOND, the split of quality into levels, which allows defining some metrics like delivery ratio and delay on each quality level independently. The set of all metrics among the different levels contribute to define an overall viewing experience value (more on Chapter 4).

To this end we propose to use the Scalable Video Coding (SVC), a state-of-the-art coding technique with an available official reference implementation (JSVM [2]). Lastly, the quality of the received videos is measured using the Peak Signal-to-Noise Ratio (PSNR) metric.

* * *

Chapter 3

RELATED LITERATURE

This chapter discusses works in the literature related to our research. It starts in Section 3.1 with a general overview of state-of-the-art video communication schemes in multihop networks. Then Sections 3.2, 3.3 and 3.4 focus more on the works that are directly related to the research problems identified in Chapter 1, namely:

- P_1 Video granulation and specific viewing QoE improvements,
- P_2 Adaptive size tuning of the video chunks, and
- P_3 Loss concealment.

Moreover, these later sections also discuss the pros and cons of the presented works with regard to their suitability for the given problems.

3.1 VIDEO COMMUNICATION SCHEMES IN MULTIHOP NETWORKS

Video applications found their way to both Delay Tolerant Networks (DTNs) and Opportunistic Networks (OppNets) through the less strict Mobile Ad-hoc Networks (MANETs). Hence, this section reviews video communication schemes in each of the given multihop domains (MANETs, DTNs and OppNets), and analyzes them based on three criteria:

1. Target multihop network.
2. Solving approach.
3. Evaluation metrics.

Each of the following subsections addresses the corresponding works from the perspective of one criteria at a time. All together then they serve to reflect the contribution, challenges and objectives of the addressed works, and analyze their visions in the context of our research.

3.1.1 *Target Multihop Network*

This subsection addresses the state-of-the-art works (that are summarized later in Table 3.1) from the perspective of their target multihop network.
MANET is the most favorable multihop network for video communication, since the presence of an end-to-end connectivity enables supporting streaming applications with strict time constraints. A number of works examine many streaming scenarios of different communication models with regard to source and destination, i.e. 1:1, 1:n, n:1 and n:n. Whereas Yu et al. [72] study a routing protocol for 1:n tree multicasting, Schierl et al. [73, 74, 75] aim to merge multiple streams of the same video from different sources by implementing a multi-source (n:1) streaming scheme[1]. Seferoglu and Markopoulou [76, 77] extend this by merging multiple streams on paths to different recipients, so that to support n:n streaming. One the other hand, Qin and Zimmermann [78] and Mao et al. [79] also want to support multiple streams on multiple paths, but on the basis of a 1:1 streaming problem.
A MANET is established by definition considering a source, a destination and a set of intermediary relay nodes, without further external entities. However, many works deviate from the standard definition, shifting their multihop network into a new dimension. For example, Seferoglu and Markopoulou [76, 77] further add an Access Point (AP) to Internet, to shift the wireless environment to a mesh network with both infrastructure and infrastructureless alternatives. Another shift of MANET by Cabrero et al. [80] consider video streaming in sparse network configurations. The network can switch between MANET and DTN depending on path availability. The authors introduce session nodes, which act as ordinary relays in the normal case of MANETs, where an end-to-end path exist. If a disconnection happens, MANETs would usually fail. However, session nodes are used then as message ferries, to deliver data between network's partitions on top of the Store-Carry-Forward (SCF) paradigm. Similarly, Raffelsberger and Hellwagner [81] assume that in emergency situations the established MANETs might suffer link breaks, leading to partition the network. Hence, they aim for the delivery of audiovisual content at switching between MANET and DTN.
In all of the works in DTNs and OppNets, streaming applications can not be supported due to the coarse conditions. Instead, data forwarding schemes are introduced, where delivery delays are unpredictable compared to the streaming counterparts. For instance, in an OppNet with indirect end-to-end delivery between individual nodes instead of network partitions, Chen et al. [82, 83] and Wu and Ma [84] discuss video data delivery over disconnections and intermittent contacts. Besides, similar to the shift of MANET to mesh networks in [76, 77], Yoon et al. [85], Yoon and Kim [86] also modify the standard definition of an OppNet for their streaming protocol. They propose the MOVi protocol for on-demand video streaming in a mobile OppNets, which

1 The solving approaches of all the listed works are discussed in details next in Section 3.1.2

is built on a Peer-to-peer (P2P) network with distributed Access Points and a central scheduler. Mobile peers always address their video requests to the central server, and a list of potential mobile providers is returned.

CONCLUSION The majority of the works in the MANET domain are neither comparable nor directly derivable in the case of very sparse networks with missing end-to-end paths and high disruptions. To the best of our knowledge, beside the deviated definition by Yoon et al. [85], the only video schemes in a correctly defined OppNet are the two by Chen et al. [82, 83] and Wu and Ma [84], in addition to the scheme under DTN by Cabrero et al. [80] and Raffelsberger and Hellwagner [81]. A classification of the reviewed schemes according to the network type, is given in Figure 3.1.

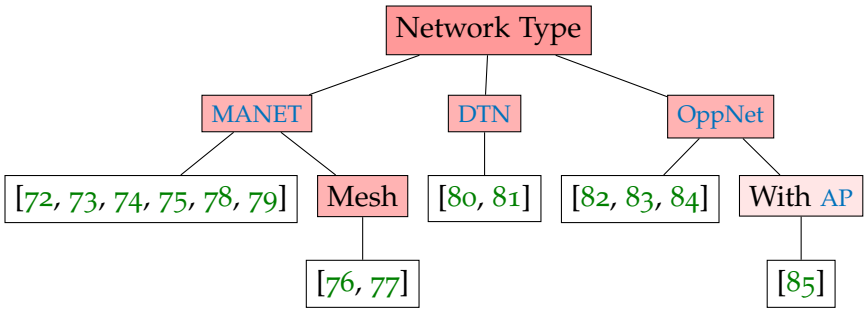


Figure 3.1: Classified Related Works: Network Types

3.1.2 *Solving Approach*

This subsection addresses the corresponding works (that are summarized later in Table 3.1) from the perspective of the proposed solving approach.

3.1.2.1 *Coding Techniques*

The most common approach for video delivery in multihop networks is to leverage the coding theory, under one of its types; network, channel or video (aka source) coding. The basic idea behind network and channel coding is to combine the original data parts into another form of packet including redundancy, to enhance throughput or help in loss recovery. Source coding, on the other hand, aims basically at data compression as introduced in Section 2.3.

NETWORK CODING It is applied in [72, 76, 77], where different network packets are combined together to enhance the communication efficiency. Yu et al. [72] use network coding to encode multicast packets of on-demand videos before being delivered. Seferoglu and Markopoulou [76, 77] enable intermediary nodes to combine

streams from multiple sources using XOR-based network codes, in order to form a single output with an increased amount of information per transmission.

CHANNEL CODING aka Forward Error Correction (FEC). It helps to detect and possibly recover errors and losses. Schierl et al. [73, 74, 75] combine both channel and video coding techniques to present a multi-source streaming approach, that overcomes video failures caused by route losses. The video, which resides at different sources (acting as servers) at the same time, is divided into layers using Scalable Video Coding (SVC). Then a FEC technique is applied to generate different correction codes per source node and SVC layer. These independent representations of the same video are routed then on demand, over different paths through the moving intermediary nodes to the requesting (or subscribed) destination(s). The different sources do not use for this any central coordinator or direct connection between each other. To decode a layer, a destination needs a subset of the originally generated codes of that layer. Chen et al. [82, 83] assume that a video is completely available at one source. Then, similarly to before, it is divided using channel and video coding into quality layers and coded blocks. Using a controlled flooding routing with a prefixed coding replication factor, the authors ensure a fixed amount of overhead. Again using SVC, Qin and Zimmermann [78] encode videos into two layers, and predict link-breaks based on a mathematical model in relation with movement patterns and directions. Using these predictions, and by monitoring the receiver's buffer, transmission bit-rate is adaptively reduced and streaming can be limited to the (so-considered) more important parts.
As seen in Section 2.3.1, Multiple Description Coding (MDC) is another used multistream video coding. To overcome frequent loss because of the multihop data delivery in MANETs, Mao et al. [79] propose multistream video coding with path diversity. A number of paths is established between the source and the destination, and each path is evaluated with Quality-of-Service (QoS) parameters like delay and loss probabilities. Videos are coded into multistreams using both layered coding and MDC. The layered coding's unequally important base- and enhancement layers are sent on different paths. Combined with a feedback channel and a retransmission-request technique, the Base Layer (BL) can be retransmitted on a path that was originally used to transmit a less important enhancement layer. On the other hand, MDC depends on prediction from previously coded frames, with no need for feedback. Raffelsberger and Hellwagner [81] manually encode videos at different specifications in order to achieve multistream videos. They divide the content into temporal segments with short durations. Furthermore, each segment is separately encoded into different representations at a variety of bitrates. Segments are prioritized based on time and representation, and a transmission order is prepared in accordance.
However, the main drawback of the coding techniques is that the source node has to

spend more time distributing the code blocks to a larger set of relays. Furthermore, when channel coding is combined with video coding, changes in the video encoder body are needed.

3.1.2.2 *Other Techniques*

Beside coding techniques, Wu and Ma [84] aim at an improved video delivery quality using a buffer management strategy, based on video properties. Video frames are split bit-wise into several messages prior to transmission. Messages are differentiated according to the type of frame, to which they belong in their GoP, that is I-, B- or P-frame message. This property is utilized to trade-off messages when deciding on which should be kept in the buffer. Another bit-wise fragmentation is applied by Yoon et al. [85] based on their P2P scheme. Before distribution, video content is fragmented like in Bittorrent into segments with identical sizes and in a certain order. And upon reception, missing parts are complemented by the central server in order to meet the strict delay constraints.

CONCLUSION Data partitioning is implicitly expressed in network and channel coding, since data composition is applied on network packets. However, both network and channel coding suffer from prolonged delays, content encapsulation and obligations of video encoder changes. On the other hand, video coding techniques help not only to overcome the bad QoS conditions, but also provide a new perspective to enhance the Quality-of-Experience (QoE) through the adaptivity and scalability options. A classification of the reviewed schemes according to the solution key, is given in Figure 3.2.

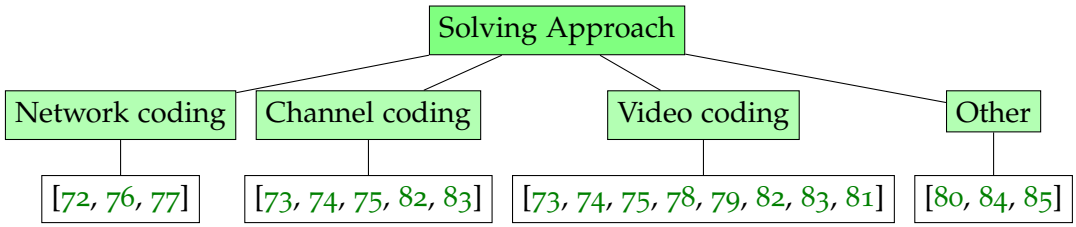


Figure 3.2: Classified Related Works: Solution Method

3.1.3 *Evaluation Metric*

It is obvious for the quality to be the dominant metric in evaluating video communication schemes. In all the works of [76, 77, 73, 74, 75, 79, 82, 83, 84], Peak Signal-to-Noise Ratio (PSNR) is used as the quality metric. Differently, Yu et al. [72] measure video playback quality by the number of successfully decoded video packets compared to their total number.

Beside quality, the delivery or loss ratios are also common metrics, since they directly affect the quality level. Loss is measured in [76, 77, 79], and delivery in [78, 80, 81]. However, this can be implemented in a different way in the context of videos. An example is the concept of playout availability by Schierl et al. [73], which considers the cases where delivery is enough to play any degraded version of the video.
Despite the importance of the delivery delay metric, it is rarely evaluated in streaming application, because it is considered as a knockout factor, that is either met or not. However, Raffelsberger and Hellwagner [81] and Yu et al. [72] measure the delay as it is, whereas Cabrero et al. [80] and Chen et al. [82, 83] take it as a function for the increasing delivery ratio and quality level respectively. A special case of the delay can be derived out of this, namely the playout delay, which accepts different reception instances at different qualities and delays. Playout delay is also used by Schierl et al. [75] and Chen et al. [82].
Otherwise, other specific evaluation metrics are introduced like the throughput and power consumption [76, 77, 85]. However, These are important metrics, and have their own research domains outside the video context.

CONCLUSION The majority of the works consider the received video quality as the main evaluation metric. Many works also use the delivery and loss ratios as well. Furthermore, although streaming applications in MANET require strict delay constraints, and thus consider delay as a knockout factor, in more constrained multihop networks network, the varying delay value is an important metric. Besides, few video schemes concentrate on other specific metrics like the throughput (network utilization) and energy consumption. A classification of the reviewed schemes according to the evaluation metrics, is given in Figure 3.3.

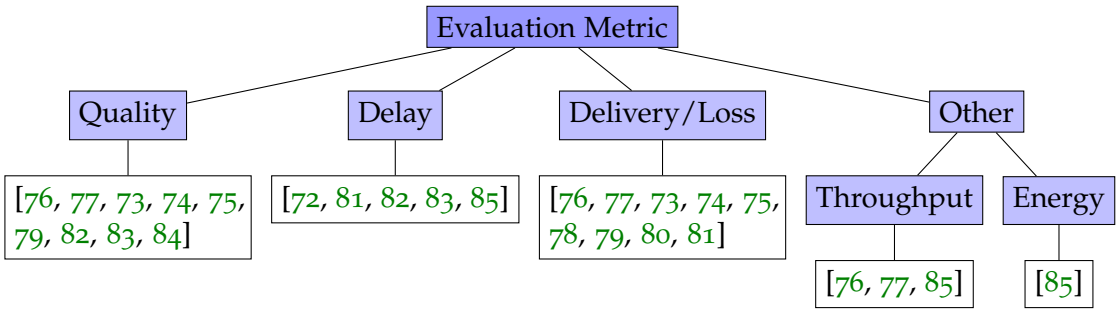


Figure 3.3: Classified Related Works: Evaluation Metrics

3.1.4 *Summary*

Summaries of all video communication schemes are given, along their characteristics in Table 3.1, and simulation setup in Table 3.2.

Work Ref.	Network	Com. Model	Solution Key	Routing	Evaluation Metric
Yu et al. [72]	MANET	Multicast	Network coding	On-demand tree-based multicasting	Multicast video playback quality, Playout delay
Seferoglu and Markopoulou [76, 77]	Mesh Network with Internet APs	Multi-source streaming	Network coding	1-hop direct delivery	Quality (PSNR) to loss, Throughput to loss
Schierl et al. [73, 74, 75]	MANET	Multi-source streaming	SVC + Raptor-codes FEC	Reactive on demand	Playout availability, Quality, Startup delay (in [75]) (PSNR)
Qin and Zimmermann [78]	MANET	Unicast streaming	SVC	Flooding with link breaks prediction	Delivery ratio
Mao et al. [79]	MANET	Unicast streaming	Layered coding, Multiple Description Coding	On demand source-based routing, over multiple paths	Quality (PSNR), Loss ratio
Cabrero et al. [80]	Sparse MANET (DTN)	Unicast streaming	Overlay DTN using Store-Carry-Forward	Optimized Link State Routing	Delivery ratio over Time
Raffelsberger and Hellwagner [81]	DTN	Unicast forward	Temporal segments + MPEG-2 different representations	Epidemic, PRoPHET and SnW	Video delivery ratio and delay of two quality levels
Chen et al. [82, 83]	OppNet	Unicast forward	Multiple Description Coding + Layered coding	Controlled-flooding	Quality (PSNR) to delay
Wu and Ma [84]	OppNet	Unicast forwarding	Bit-wise frame split, priority based on GoP frame type	Epidemic	Quality (PSNR) to buffer tunning parameters
Yoon et al. [85]	OppNet with APs and a central server	Peer-to-peer (P2P) streaming	Bittorrent-like fragmentation	P2P with central-scheduler	Playout delay and continuity, Throughput, Power consumption

Table 3.1: Literature Summary

Work Ref.	Simulator	Mobility	Environment	Num. of Nodes
Yu et al. [72]	Self implemented	RWP	Open area: 1000×1000m	60
Seferoglu and Markopoulou [76, 77]	NA	None	NA	NA
Schierl et al. [73, 74, 75]	NS-2	RWP	Open area: 1000×600m	40
Qin and Zimmermann [78]	NS-2	RWP, RW	NA	NA
Mao et al. [79]	OPNET	RWP	Open area: 600×600m	16
Cabrero et al. [80]	Emulator (compatible with NS-2)	RWP	Open area: 1000×600m $\rightarrow$ 3000×1000m	$2-20 \rightarrow 30-50$
Raffelsberger and Hellwagner [81]	ONE	Real Mobility-Driven: Disaster area	Real map-based: 400×300m	25
Chen et al. [82, 83]	DTNSim	Real-world traces	NA	273, 5148
Wu and Ma [84]	NS-2	RWP, Real-world traces	Open area: 1000×1000m, Traces-based area: 10×10km	60, 92
Yoon et al. [85]	NS-2	Real-world traces	Traces-based area: 1000×1000m	500

Table 3.2: Literature Summary, Simulation Setup

3.2 GRANULATION AND CORRESPONDING ROUTING

This section discusses related works with regard to the first research problem, as defined in Chapter 1 :
(P_1) *How to granulate a video and route the resulting chunks, to improve the viewing QoE and resist partial losses?*

3.2.1 *Granulation and Data Partitioning*

It has been seen in Section 3.1 that when dealing with large volumes of data in multihop networks, granulation exists as a part of the solution in the majority of the works. This manifests itself in one or a combination of the following forms:

1. File-level partitioning
2. Network coding
3. Channel coding
4. Video coding

This step is fundamental to prevent limited contacts times, which are the common case in OppNet, from frustrating large data delivery [87, 88].

FILE-LEVEL PARTITIONING This is interpreted as dividing the data into parts[2] at the physical file level. Pitkanen et al. [88] study how to partition the data payload, where some fixed information of the original header (e.g. source, destination) have to be copied as it is into each of the parts, and other headers (e.g. size, partitioning offset and routing information) have to be updated accordingly to enable routing and recovering the original data. Recovery can succeed only if all the composing parts are received.
In the context of videos, Wu and Ma [84] divide each video frame into several parts, and differentiate between them according to their originating frame. They rely on the dependency between I-, B- and P-frames within a GoP to prioritize data parts. Yoon et al. [85] build a bittorrent-like P2P sharing protocol, and assume hence that data is divided into segments, as this kind of protocols require. For a different purpose, Renuka and Thangaraj [89] and Vinod and Madhusudan [90] divide data into smaller parts to enhance transmission confidentiality. Each part is encrypted separately, and then using multipath routing, an attacker has a lower probability to reconstruct the original data.
Despite simplicity, this kind of partitioning may suffer bad worst-case delays if one

2 Can also be named chunks, fragments, granules or partitions.

(or more) of the composing parts arrive late. Furthermore, without further protection (e.g. redundancy), any loss can cause a delivery failure.

NETWORK/CHANNEL CODING The basic idea behind network and channel coding is to take the original data parts, combine them into another form of packets that are called code blocks, and include extra blocks to help in loss recovery. At the receiver side, only a subset of the total blocks is needed in order to reconstruct the original data. Network coding is used by Wang et al. [91] and Tsapeli and Tsaoussidis [92] to further enhance routing in OppNets. Moreover, as seen in Section 3.1, network coding is also used for multicasting [72, 76, 77] and channel coding for streaming [73, 74, 75] in MANETs.
The main drawback of these techniques is that the source node has to spend more time generating and distributing the code blocks to a larger set of relays. Furthermore, when channel coding is combined with video coding, changes in the video encoder body are needed.

VIDEO CODING An overview of video coding is already given in Section 2.3, and its common usage in Section 3.1. The scalable granularity implemented in this approach can also be combined with channel or erasure coding to support more error resilience [73, 74, 75].

3.2.2 *Routing of The Data Parts*

Based on the selection of Scalable Video Coding (SVC) as given in Section 2.3.2, we have to consider that the resulting data parts are unequally important due to their dependency. Consequently, the communication scheme to transmit these parts must respect the different importance levels, in order to meet the viewing experience requirements. This philosophy of inequality can be applied in two forms: prior to routing, and on routing.

PRIOR TO ROUTING Techniques of this type apply unequal error protection functions, based on network or channel coding. Different parts are protected differently by tunning the redundancy variable of the coding algorithm, according to their importance [93, 94, 73].

ON ROUTING Techniques that function on routing can also be classified into two types:

- Drop-based: An intelligent selection/drop procedure is performed at intermediary nodes along the path, to best enhance the quality gain [76] or manage the buffer occupation [32, 84].

- Path diversity: When the sender has a diversity of paths, they can be evaluated according to some parameters, and the best or more stable path is assigned for the important data [79]. Another possibility to compare paths, is to send the important data redundant on more than one path [95].

3.2.3 *Conclusion*

This section discussed the popularity of data partitioning approaches for large data delivery in multihop networks. A summary of the corresponding works is given in Table 3.3.

Technique	Used in	Pros	Cons
File-level	[84, 85, 89, 90]	Simplicity, Parts can take any size	Bad worst-case delay, High loss probability
Network/Channel Coding	[91, 92, 72, 76, 77, 73, 74, 75]	Error resilience, Loss tolerance	Prolonged delay, Changes in the video encoder
Video Coding	[73, 74, 75, 78, 79, 82, 83]	Scalability, Content-based representation of sub parts, Loss tolerance	Configuration complexity

Table 3.3: Granularity Techniques

Then, based on the arguments in Section 2.3.2 for choosing SVC, the applied routing protocol has to respect the fact the resulting data parts are not similarly important. Hence, unequal routing protocols are presented.
Our contribution in this context, and our positioning with regard to the presented works, are discussed later in Chapter 4.

* * *

3.3 ADAPTIVE SIZE TUNING OF THE VIDEO CHUNKS

This section discusses related works with regard to the second research problem, as defined in Chapter 1 :
(P_2) *How to adaptively tune the size of the data chunks, with respect to OppNet's changing conditions?*

Adaptivity corresponds to the ability of nodes to react to their changing conditions in an OppNet: density, mobility pattern and network connectivity status. According to Moreira and Mendes [96], these critical condition changes can be basically captured by three parameters :

- Contact time: represents how long a contact between two nodes lasts.
- Inter-contact time: is the duration between two successive contacts
- Contact volume: reflects the quality of the contact that allows to transfer a set of information.

The density of the network, and how the nodes move, directly affect the contact and inter-contact times. How the nodes react then, belongs to one of two forms at two levels of the routing protocol: adapt the message(s) to be sent, or adapt the routing decision (to send or not, and to whom to send).

3.3.1 *Message Adaptation*

This form of adaptation answers the question: what to send? It consists most commonly in reducing the data rate according to the connectivity status. By using network or channel coding, the replication factor to generate the coding blocks can be adapted according to how good the connectivity is, in order to enhance delivery or reduce overhead [97, 75, 74].
Another way of adapting what to send, is how Tournoux et al. [98] optimize the number of copies omitted by Spray-and-Wait routing protocol to enhance delivery under delay constraints and keep the overhead as low as possible. They used in this context the node degree parameter, which was defined by them as the average number of contacts that a node encounters in a given period of time.

3.3.2 *Routing Adaptation*

A wider space of adaptation possibilities, is to adapt how routing protocols function, which answers the two questions: If and How to send? One of the first examples in this context are the PRoPHET [33] and MaxProp [99] routing protocols. Based on the contact history, MaxProp prioritizes the messages to be sent or dropped, whereas PRoPHET forwards a message only to those nodes that have a higher delivery probability.
Similarly, Hu and Hsieh [100] provide each node with a limited list to record the history inter-contact times. This list helps to measure the current local density and estimate the future one. Consequently, nodes can take density-aware decisions to forward messages only to encountered nodes that are moving toward a dense area.
Wang et al. [32] modify the epidemic routing protocol to make message forwarding when encountering a node subject to variable weighting and probabilistic functions. The forwarding probability function is evaluated through the message's level of pervasiveness among neighbor nodes, which is defined as "the proportion of nodes encountered that have the specific message over a predefine time period [32]". Moreover,

which messages to be forwarded is determined my the messages' weights, pervasiveness as well as an inter-contact time measure.
The routing decision is not limited to whether to send a message or not, but can also serve to choose routing protocol itself. For instance Lakkakorpi et al. [101] enable the sending node to adaptively switch the routing protocol between MANET-based and DTN-based[3]. This decision is taken depending on a set of parameters: node density, which inferred from tracked contact pattern, and velocity and file size, which both can be directly read as local information.
Miao et al. [102] rely on two parameters to estimate the delay and cost of data delivery: centrality and regularity, and hence switch between two corresponding routing algorithms in accordance. The former parameter is the node's relative importance to other nodes (measured by the number of times that node was on the path of a message that was delivered to them), and the latter is the likeliness for two nodes to come into contact based on their encounter history.

3.3.3 *Conclusion*

A summary of all mentioned works is given in Table 3.4.

Adaptation	Work Ref.	Measurement	Reaction
Message	[97, 75, 74]	Connectivity status	Change messages' redundancy factor
	[98]	Average number of contacts	Optimize SnW's number of copies
Routing decision	[33]	Contact (encounter) history	Forward or not
	[99]	Contact (encounter) history	Prioritize messages
	[100]	Inter-contact time	Forward or not
	[32]	Message pervasiveness & Inter-contact time	Forward or not
	[101]	Number of contacts & Nodes' velocity	Switch routing protocol
	[102]	Contact (encounter) history	Switch routing protocol

Table 3.4: Related Works Summary: Adaptivity

3 Differences are presented in Section 2.1.1

3.4 LOSS RECOVERY

This section discusses related works from the perspective of the third research problem, as defined in Chapter 1 :
(P3) *How to design a loss concealment mechanism at the destination node, to enable reacting to small amounts of loss?*

There are different terms that refer to overcoming loss, namely *resilience*, *concealment* and *recovery*. These terms are differentiated depending on how missing data are compensated.

RESILIENCE reconstructs lost (or faulty) packets using redundant codes that are extracted from the original data and added by the transmitter [103]. This is similar to how network and channel coding work, as presented in Section 3.2.

CONCEALMENT tries to reconstruct lost data using the already received parts of the original data [103]. For example, a missing video frame can be reconstructed using the adjacent frames if they were received.

RECOVERY is a general term, which includes the last two ones. Moreover, it proposes to implement other generic techniques like packet-retransmission [104].

In the context of this dissertation, loss recovery is approached through two perspectives: video frame loss concealment, and network packet loss recovery.

3.4.1 *Video Frame Loss Concealment*

Videos are characterized by high dependency among the underlying components, e.g. consecutive frames, or blocks within the same frame. This property helps to overcome an amount of loss, using the successfully received parts. However, there is no direct contribution in this work to this kind of concealment. Therefore, we will only give a quick review of existing techniques and corresponding surveys, in order to enlighten our selection of applied techniques.
Concerns of video loss concealment in SVC go back to its predecessor H.264/Advanced Video Coding (AVC) [105]. Although the new capabilities and characteristics of SVC increase the encoding complexity, SVC still has a plenty of concealment techniques, most of which are inherited from the error concealment literature of the AVC. Guo et al. [103] describe the transition from AVC to SVC, and outline the key differences between the concealment techniques of each of the coding techniques. Based on the layered structure of SVC, they classify the techniques into *Intra-layer*, where the concealment information are taken from the same scalability layer, and *Inter-layer*, where the concealment information is exchanged among layers. The SVC Joint Scalable Video

Model (JSVM)[4] reference software implementation includes four built-in basic techniques:

1. Frame Copy: the last received reference frame is entirely copied into the location of the missing frame.
2. Temporal Direct: only the motion vectors of the last reference frame are copied, and then scaled to fit the place of loss.
3. BLSkip: upsamples the motion vector and residual information from the (correctly received) BL to fit the lost spatial layer.
4. Reconstruction BL Upsampling: decodes the BL directly and upsamples the textural elements.

Where the techniques (1-2) function in an intra-layer manner, and (3-4) in an inter-layer one. Two surveys and extended evaluations of the above mentioned concealment techniques are given in [106, 107].

3.4.2 *Network Packet Loss Recovery*

To the best of our knowledge, loss recovery in OppNets is mainly addressed through the resilience perspective, which usually follows network and channel coding algorithms [91, 92]. The basic idea, as seen in Section 3.2, is to include extra code blocks to the original data body, and these will be transmitted through a big number of relays [108], usually bigger than the required number of relays for ordinary replication-based routing protocols. At the receiver side, only a subset of the blocks is needed in order to reconstruct the original data. The main drawback of these techniques is that the source node has to spend more time distributing the code blocks to a larger set of relays, and consequently a prolonged delay is observed. Multisourcing was suggested as a solution to overcome this problem [109], where code blocks are initiated by more than one source node. Nevertheless, the multisourcing technique is tightly dependent on the coordination possibility between the sources for the generation of data and their code blocks, and such a coordination is not always possible according to the network settings.
Network and channel coding are also specifically applied for SVC applications, using Raptor erasure coding [73, 110] or Reed-Solomon coding [111]. In addition to the above mentioned drawbacks, using these coding techniques for SVC requires changes in the encoder and/or byte-wise partitioning to build the code blocks, which contradicts the advantages of Network Abstraction Layer (NAL) units partitioning.

4 Version: 9.19.14 - Last update: 14.06.2011

3.4.3 *Conclusion*

A summary of all mentioned works is given in Table 3.5.

Technique		Work Ref.	Pros	Cons
Video loss concealment		[105, 103, 106, 107]	Rely on existing data	High impact on quality, Advanced techniques are concerned with high computation complexity
Recovery	General com. schemes	[91, 92, 108, 109]	Reliability, Computationally simple	Prolonged delay OR Coordination necessity
	Video com. schemes	[73, 110, 111]	Reliability, Computationally simple	Encoder changes

Table 3.5: Related Works Summary: Loss Recovery

* * *

Part II

CONTRIBUTION

Chapter 4: Toward Video Delivery With a Better Viewing Experience Using SVC: An Experimental Study.

Chapter 5: Adapting Granularity for A Better Contact-Opportunity Exploitation.

Chapter 6: The Integration of BALCON: A Backward Loss Concealment Mechanism.

Delay is preferable to error.

— Thomas Jefferson

Chapter 4

TOWARD VIDEO DELIVERY WITH A BETTER VIEWING EXPERIENCE USING SVC: AN EXPERIMENTAL STUDY

4.1 OVERVIEW

This chapter addresses the first research problem, which focuses on video granulation to shift the transmission units from entire videos to smaller parts. It also addresses the viewing-experience measurement, which is defined then as a compromise between Quality-of-Service (QoS) and Quality-of-Experience (QoE). Consequently, the communication scheme will aim at optimizing the delivery results considering delivery delay, overhead and video quality.
Toward these objectives, this chapter proposes an experimental study that covers the selection of the granulation mechanism and the routing of the resulting data parts. The configuration parameters for both video encoding and network simulation are studied, and experimental results are derived and discussed.

4.2 OBJECTIVES AND CONTRIBUTIONS

This section highlights the objectives and contributions of this chapter. First, the need for granulation and the importance of choosing Scalable Video Coding (SVC) is addressed. Then, we define the notion of viewing experience and how it is measured in our context benefiting from the characteristics of SVC. Lastly, with respect to this definition, a prioritized redundancy-based routing is formulated.

4.2.1 *Recall: Granulation and Data Partitioning*

In Chapter 2, Opportunistic Networks (OppNets) were presented as human-centric mobile ad-hoc networks, with no constraints on mobility, stable connectivity nor topology changes. Nodes' free mobility widens the network topology, and provides

indirect end-to-end data delivery on top of the Store-Carry-Forward (SCF) paradigm. However, that limits data links to be operational only during opportunistic contacts, which might not last for long durations. Consequently, typical applications in this context usually consider only small messages, which can easily fit with the short contact times, as for instance disaster alarms and traffic notifications.
To enable applications of large data volumes, granulation emerges as a necessary step to break large messages into smaller sub-messages, which can exploit short opportunistic contacts. At the receiver side, recovery of original data succeeds only if all sub-messages are successfully received. This reflects a high sensitivity toward loss, so that any partial loss can cause a high delivery delay or a whole failure. A temporary loss until the late arriving one or more sub-messages causes a high delivery delay, that is associated with the last arriving sub-message. If loss turns permanent, with regard to a time constraint, then delivery is failed. Delivery is usually enhanced by applying different forms of redundancy to avoid partial losses. However, redundancy is associated with extra overhead in the network, which actually identifies the problem as a trade-off between the QoS parameters of delay, overhead and delivery probability.
Unlike generic content delivery, videos can alleviate the sensitivity to loss by tolerating an amount of loss at the cost of a lower quality. In this context, SVC does not only comprise the core idea of granulation by dividing the video main stream into smaller sub-streams, but it also introduces the advantages of quality scalability. As described in Section 2.3.2, SVC encodes a video into one Base Layer (BL) and several Enhancement Layers (ELs). The first layer represents the video at its lowest quality specifications, and each further layer then enhances the quality in one of the scalability dimensions. This property is exploited below to formulate an evaluation metric of the viewing experience.

4.2.2 *Viewing Experience*

As outlined before, the QoS covers technical network performance indicators, whereas the QoE goes further in video applications and measures the satisfaction of end users with the perceived video quality [63, 64].
Thanks to using SVC, quality is divided into levels, and can be regarded as a further trade-off dimension in addition to delay, overhead and delivery probability. This allows to extend the QoS model to represent QoE as well. The quality level is tightly dependent on the delivery ratio on a multi-level scale. That is, independent delivery ratio measurements must be taken for each corresponding quality level. Consequently, in video applications, the metric of quality level can substitute its delivery ratio counterpart, and hence both will be interpreted as a single parameter. An overview of the trade-off parameters for generic and video data delivery is depicted in Figure 4.1.

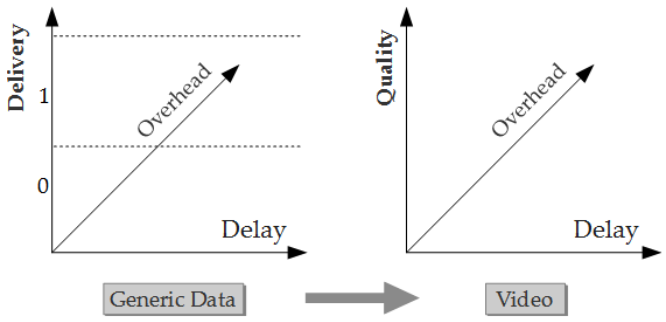


Figure 4.1: Trade-off Dimensions for Generic and Video Data Delivery

The objective of the communication scheme in our work is therefore to optimize results in the depicted 3D space, with respect to the parameters: quality, delay and overhead.

WHERE IS THE OPTIMAL POINT IN THIS SPACE? Both delay and overhead reflect a bad performance when their values increase. Hence, their optimal values are at zero. The quality is contrarily better with higher values. Therefore, to subject a direct proportion between the three parameters, the optimal point in the trade-off space can be moved to the $(0,0,0)$ center by inverting the quality metric as shown in Figure 4.2.

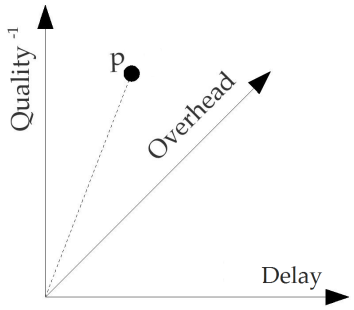


Figure 4.2: Magnitude

An experiment of sending a video under specific settings is concluded by the means of three measurement values: quality, delay and overhead. Observing these measurements as three features in a 3D feature space allows us to represent the experiment by a feature vector that is given by a point p in the 3D space in Figure 4.2. Consequently, the given experiment can be assigned a single value by simply using the Euclidean distance (magnitude) of its feature vector, which is calculated as the following:

$$val(p) = |\overrightarrow{0p}| = \sqrt{w_1.q^2 + w_2.t^2 + w_3.o^2} \tag{4.1}$$

where

q : Projection of p on the inversed quality axis

t : Projection of p on the delay axis

o : Projection of p on the overhead axis

$w_{1,2,3}$: Weights of the corresponding parameters

NORMALIZATION However, since the values of the three features belong to different data ranges, the calculated Euclidean distance will be dominated by the feature of the broadest data range. Therefore, each feature value has to be independently normalized to the $[0,1]$ range. This is done using the technique of linear scaling to unit range [112], which is calculated for a given feature value (measurement) as the following:

$$\tilde{x} = \frac{x - l}{u - l} \tag{4.2}$$

where

$\tilde{x}$: The normalized value of x, where $\tilde{x} \in [0,1]$

l : The lower bound of the corresponding feature's data range.

u : The upper bound of the corresponding feature's data range.

By applying this normalization to all the q, t and o components of Equation 4.1, the given formula will enable comparing different experiments using their corresponding projected points in the trade-off space. A smaller Euclidean distance refers to a better experiment, because the feature vector will be closer to the $(0,0,0)$ center, where the optimal point lies.

WEIGHTING The weighting factors are assigned depending on the target application, if any of parameters is more critical than the others. For example, in video telephony, the delay is very critical to ensure real-time conversations, then w_2 gets a higher value than w_1 and w_3 to express this. On the other hand, in movies streaming, some users are interested in a very good quality, then w_1 must get a higher value. However, when not comparing experiments form different applications the weighting can be ignored by assigning $w_1 = w_2 = w_3 = 1$.

THE INFLUENCE OF MULTI-LEVEL QUALITY In the multi-level quality context of SVC, we follow the argumentation by Schierl et al. [75] and Chen et al. [82], which states that a lower playout delay at the cost of a lower quality would ensure a better viewing experience. Therefore, in our further experiments, we propose to extend the single result point in Figure 4.2 to two points that can represent the result of an experiment:

p_1: represents the result associated with the lower playout delay at the SVC's BL quality.

p_2: represents the result associated with the maximum delivered layer. At the worst case, if no further ELs beyond the BL could be delivered, p_2 would be identical to p_1.

An example of two result points in a 2D space of inverted quality and delay (referred to the axes as Q and T respectively) is depicted in Figure 4.3.

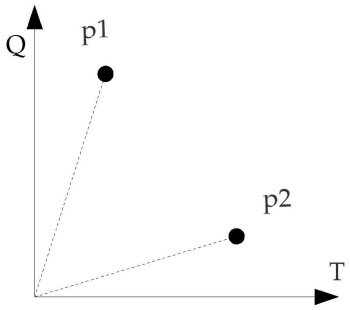


Figure 4.3: Two Result Points

To represent an experiment with a single value that takes both p_1 and p_2 into consideration, an average feature vector between p_1 and p_2 is calculated. Consequently, the viewing experience is implicitly represented by the means of the playout delay (taken from p_1), the maximum delivered level of quality (taken from p_2) and the overhead (homogeneously calculated for the experiment).

4.2.3 *Prioritized Routing*

SVC output layers, $L_0, ..., L_n$, refer to different importance degrees because of two reasons, one is related to the technical characteristics of SVC and the other one is derived from our viewing experience requirements.

FIRST, the layers of a scalable bit stream form a hierarchical dependency [54, 113]. That is, a given layer is not decodable unless all of the lower layers, on which it depends, are already successfully received. Consequently, given n scalable layers, the base layer L_0 is essential for all upper layer, and hence the most important layer. On the other hand, the last layer L_{n-1} is the least important one.

SECOND, based on the interest of delivering a degraded version of the video with the shortest possible delay for a better viewing experience, more importance must also be assigned to lower layers, which represent a video at its lower specifications. Consequently, same as before, the base layer L_0 is associated with the highest priority.

This prioritizing affects the selection and configuration of the routing protocol. As presented in Section 3.2, the choice of a routing protocol that is based on controlled flooding and multi-path redundancy encloses many advantages. First, it allows to simply reflect the priorities of messages by tuning the corresponding redundancy factor up and down. That is, one can send more copies of the more important messages (i.e. layers), and vice versa. Second, it offers a full control over the applied network overhead. The Redundancy factor, or the allowed number of copies, define an upper bound of the overhead. Lastly, when it comes to OppNets, where no prior knowledge on the network or the nodes exists, this family of routing techniques (e.g. Spray-and-Wait (SnW)) report good experimental results with regard to delivery ratio, delay and limited overhead [114].
Hence, the workflow of our proposal is based on the SVC encoding of an input video into a given number (n) of layers, over which m Operating Points (OPs) are defined as described in Section 2.3.2.2. Then, the BL is sent first with the highest redundancy factor, followed by the second important OP with less redundancy, and so on until OP_{m-1}. Upon each contact on the way to destination, messages are copied based on the algorithm of the SnW routing protocol, as described in Section 2.1.2. The workflow is summarized in Listing 4.1, and an example run of the proposition is depicted in Figure 4.4.

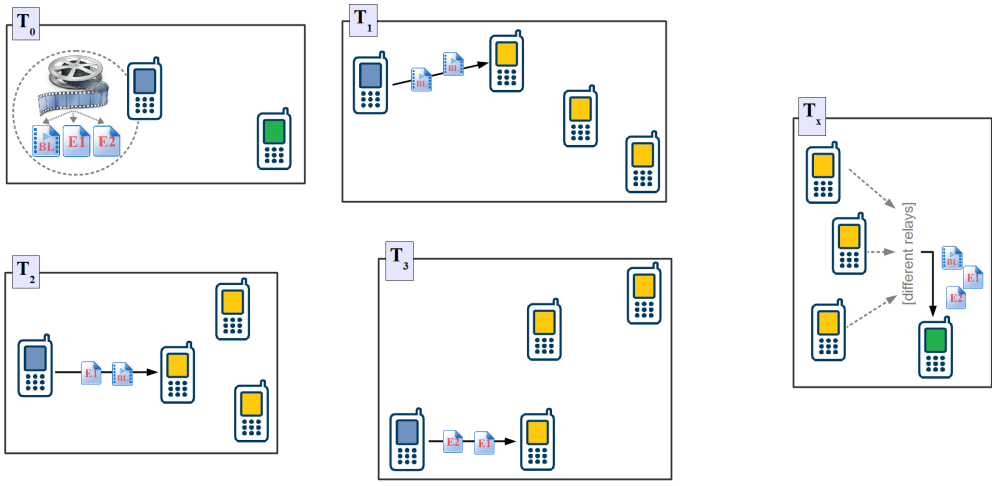


Figure 4.4: An Example Run of Prioritized Layered Routing

```
Input: raw single-stream video

% Granulating and preparing messages @source
svcVid = JSVM-SVC-encode(video) % where svcVid has n layers: L[0 .. n-1]
Define m < n operating points: OP[0 .. m-1]
For (j = 0 → m − 1) {
   OP[j].setRedundancyFactor()
   % where OP[j].redundancyFactor >= OP[j+1].redundancyFactor, because lower
       layers and OPs are more important as already given in Section 4.2.3
}

% Routing of a message msg=OP[j] when the node X meets Y
if (Y.hasMessage(msg) == false) {
   if (X.msg.redundancyFactor > 1) {
      X.send(to=Y, message=msg)
      X.msg.redundancyFactor /= 2
      Y.msg.redundancyFactor /= 2
   }
   else { % when redundancyFactor=1, the message can only be sent to its target
      if (Y == X.msg.target) {
         X.send(to=Y, message=msg)
         X.delete(msg)
      }
   }
}

% Measurements @target
t1: Delay to deliver OP[0] = BL
q1: BL quality level

t2: Delay to reach OP[j];
   where 0 < j ⩽ m − 1 and all OP[0 .. j-1] are already delivered
q2: Quality level associated with OP[j]

o: ∑_{j=0}^{m−1} OP[j].redundancyFactor x OP[j].size

% Evaluation
p1: The result point that corresponds to the lower playout delay at BL quality (
    t1 and q1)
p2: The result point that corresponds to maximum delivered quality at any delay
    (t2 and q2)

Evaluate the whole experiment with a single value = mean(p1, p2)
```

Listing 4.1: Prioritized Layered Routing

4.3 EXPERIMENTAL SETUP

4.3.1 *Scalable Video Coding*

This section describes the choice of parameters that are needed in the SVC encoder for the determination of the scalability dimensions and the corresponding scalability layers and operating points.

4.3.1.1 *Video Sequences*

Video streams are selected for the experiments from the video trace library [115]. Videos are downloaded in a raw and uncompressed file formats with (`.YUV`) extension. Basically, four video streams (which are commonly used standard video test sequences can be found) were selected for different experiments in the work. They all possess the same resolution of CIF: 352×288, the same frame rate of 30fps, but different number of frames and hence different file sizes.
These sequences are:

V_0 : *Bus,* consists of 150 frames

V_1 : *Crew,* consists of 600 frames

V_2 : *Highway,* consists of 2000 frames

V_3 : A repetition of *Highway,* consists of 4000 frames

(a) Bus (b) Crew (c) Highway

Figure 4.5: Video Sequences

The first sequence (*Bus*) is used only for pre-configuration experiments, whereas the other three sequences are used later for the proposed delivery scheme. Lengths of these sequences range between 20 seconds to about 2 minutes to coincide the given example scenarios in Chapter 1 (e.g. disaster recognition, touristic shooting or demonstration documenting videos), where no longer video samples are required to correspond to those examples.

4.3.1.2 *Scalability Dimensions*

As shown before, SVC supports three dimensions or features of scalability, and a given video can be encoded using SVC with any combination of the three dimensions. To set the scalability dimensions that our encoded sample videos will have, different combinations are experimented and compared with regard to 1) the size distribution among the layers and 2) the output file size. To the best of our knowledge, no other work in the literature has proposed such a study. Thus, we had to perform the experiments ourselves.

For the experiments, *Bus* and *Highway* video sequences were selected, as they are significantly different with regard to length (Bus: 150 frames, Highway: 2000 frames) and in term of dynamicity of the video content (Bus: very changing, Highway: less changing). Each sequence was encoded several times with a single isolated scalability dimension each time, and then all dimensions were combined together for the last experiment. When tested individually, temporal, spatial and SNR scalabilities included three scalability levels for each, to show the distribution of size among the layers. The last experiment combined all the scalability dimensions together, holding three temporal-, two spatial- and two SNR-levels, which results in 12 layers. This configuration is based on the recommendations provided in [59] and [116] for a better performance with regard to encoding time and output size.

Figures 4.6 and 4.7 show for the 2 videos the percentage of size distribution among the three composing layers of the individually tested scalability dimensions, as well as the 12 layers of the combined dimensions.

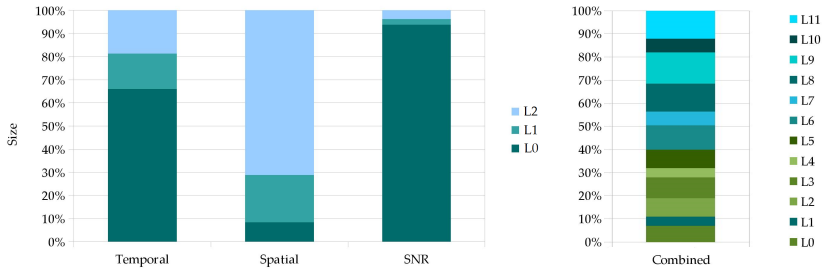


Figure 4.6: File Size Distribution Among Scalability Layers, Video: Bus

It is noticed above in the cases of the individually tested scalability dimensions, that the major part of the total size is concentrated in one layer (L_0 for the temporal and Signal-to-Noise Ratio (SNR) scalabilities, and L_2 for the spatial scalability). This contradicts the objective of fragmentation because a large sub-part is not different from the original data. On the contrary, combining all scalability options together will have the advantage of spreading the same (or only slightly larger) size over a bigger number of layers (as shown on the right of the figure).

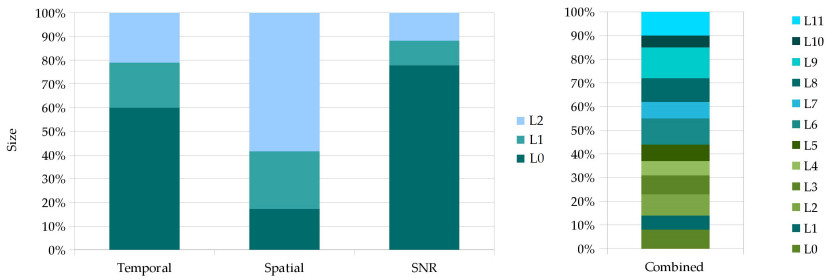


Figure 4.7: File Size Distribution Among Scalability Layers, Video: Highway

Furthermore, Figure 4.8 compares the sizes of the output files (taken for both videos), considering the encoded video with temporal scalability only as a reference, because as given in Section 2.3.2.2, any SVC-encoded video (with Group-of-Pictures (GoP)>1) is temporally scalable by default, with no extra volume or encoding complexity.

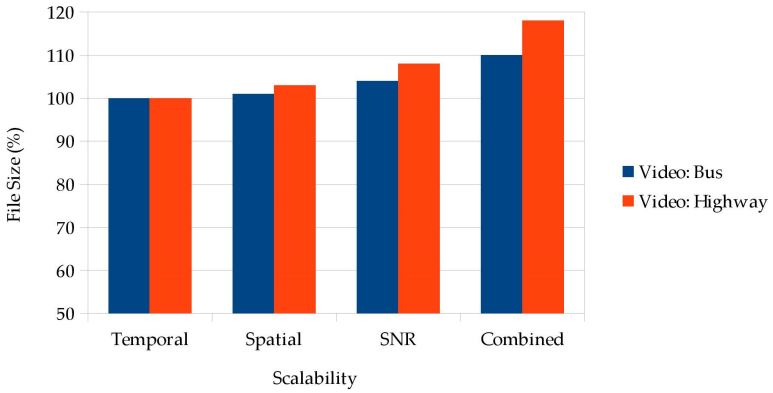


Figure 4.8: SVC File Size Comparison

This shows that encoding a video using the three scalability options yields only about 5-15% more size for the output file.
Therefore, this configuration of combined scalability dimensions will be adopted for all next experiments.

4.3.1.3 *Scalability Layers*

This section details the parameters, which define the number of layers in each scalability dimension.

TEMPORAL LAYERS As outlined in Section 2.3.2, the essential parameter to be set is the size of the GoP, which influences the coding efficiency, output file size, output quality, and more importantly the temporal scalability levels. Unanue et al. [59] listed recommended configurations for different scenarios. For a better performance with regard to encoding complexity and storage requirements, smaller GoP values are recommended. Namely, GoP is set to 4, which results in 3 temporal levels.

SPATIAL LAYERS The common resolution of video samples is the CIF (352 × 288), which allows only one step down on a 2:1 ratio due to technical limitations of the Joint Scalable Video Model (JSVM) software [2]. Two spatial resolutions is also the common recommendation by Wang et al. [116].

SNR LAYERS In relation to the recommendations by Unanue et al. [59] regarding performance oriented scenarios, they also limited the encoding to two SNR levels. The reason is that the SNR scalability is the most computationally expensive variation compared to the temporal and spatial scalabilities, and this complexity increases with more SNR layers [54].

Using these recommendations together will result in 12 scalability layers in total among the three scalability dimensions. This is illustrated in Figure 4.9.

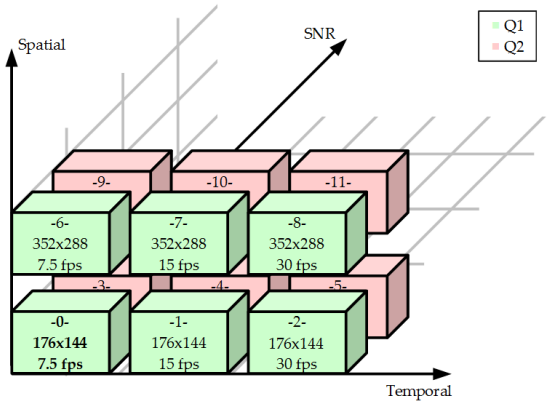


Figure 4.9: SVC Layers

4.3.1.4 *Operating Points*

The path of the OPs from the first till the last layer must be selected with respect to the following criteria:

- The path must have the maximum possible length, as given in Equation 2.2 in Section 2.3.2.
- The path must leave the SNR enhancement to the last step, because it delivers the least significant quality gain [59], and leaves the biggest peak of size to the last enhancement step.
- The path must ensure that each step starting from the first layer (BL) brings the most possible quality-gain (with regard to Peak Signal-to-Noise Ratio (PSNR) metric).

The whole selection process is summarized in Algorithm 4.2.

```
Input:
   layers: array of n layers
   paths: array of all possible OPs paths from layers[0] → layers[n − 1]

% Calculate max length according to Equation 2.2
maxLength = distanceManhattan();

% (1) filter out paths that are shorter than maxLength, and paths that enhance
    on SNR dimension before the last step.
paths = filter(paths, maxLength);

% (2) remove all paths that have very varying sizes, i.e. some OPs are too big,
    others are too small.
candidatePaths = paths
for (p in candidatePaths) {
   p.pop(lastStep); % exclude the last SNR step
   sd = stddev(sizes of operating points in p);
   if (sd > size of any operating point in p) {
      candidatePaths.pop(p);
   }
}

% if only one path is remaining after filtering, return it
if (candidatePaths.length == 1) {
   return candidatePaths[0];
}

% if more than one path is remaining, trade-off them
% if no path is remaining, trade-off remained paths after applying the filter
if (candidatePaths.length == 0) {
   candidatePaths = paths;
}

% (3) now trade-off each step from the remaining candidate paths
bestPath = [];
currentLayer = layers[0];
bestPath.push(currentLayer);
while(bestPath.length < maxLength) {
   steps = nextPossibleSteps(from=currentLayer, on=candidatePaths);
   currentLayer = chooseBestLayer(steps)
   bestPath.push(currentLayer);
}
```

Listing 4.2: The Selection of Operating Points Path

Where the trade-off between the next possible steps is done by analyzing how much quality-gain (PSNR) would the next step bring, as shown in Algorithm 4.3. The result usually depends on the video content type. That is, if the video content is very dynamic (changes a lot), the temporal enhancement brings more quality gain because it brings new different frames from previous ones. On the other hand, if the video content is stable, the spatial enhancement brings more quality gain.

```
function chooseBestLayer(stepLayers) {
   bestLayer = null;
   for (l in stepLayers) {
      if (videoPsnr(with layer l) > videoPsnr(with layer bestLayer)) {
         bestLayer = l;
      }
   }
   return bestLayer;
}
```

Listing 4.3: The Selection of Operating Points Path

Consequently, five OPs are fixed on top of the SVC layers in a way to ensure a quality advancement with each further level and achieve a good distribution of the total size among the maximum possible number of levels. Fixing the least significant SNR scalability to the last enhancement level, leaves 3 possible taxicab paths from $L_0 \rightarrow L_{11}$:

1. $L_0 \rightarrow L_6 \rightarrow L_7 \rightarrow L_8 \rightarrow L_{11}$
2. $L_0 \rightarrow L_1 \rightarrow L_7 \rightarrow L_8 \rightarrow L_{11}$
3. $L_0 \rightarrow L_1 \rightarrow L_2 \rightarrow L_8 \rightarrow L_{11}$

The distribution of size among the 5 OPs for the 3 path variations is depicted in Figure 4.10.

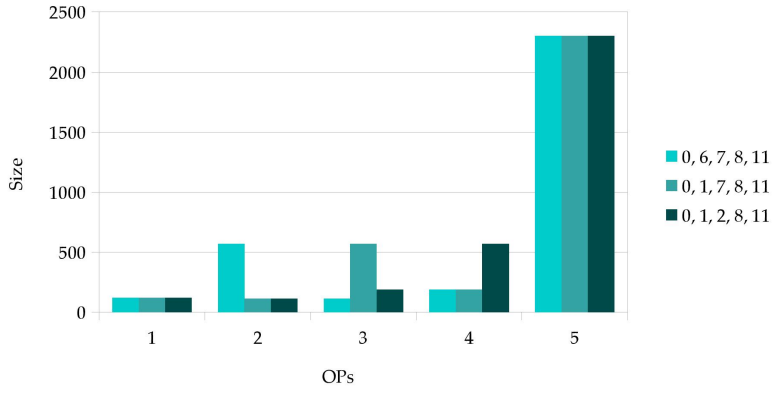


Figure 4.10: OPs Size Distribution

It is noticed that the 3 paths have the same characteristics, and there is no preferred path with regard to the size distribution. Therefore, based on the steps trade-off, we select the first path ($L_0 \rightarrow L_6 \rightarrow L_7 \rightarrow L_8 \rightarrow L_{11}$), which goes in the direction of the spatial resolution first. The spatial resolution is more important for the given example scenarios (e.g. disaster recognition, touristic shooting or demonstration documenting videos) as it critically influences the picture-quality and the needed details inside. The chosen path of OPs is shown in Figure 4.11.

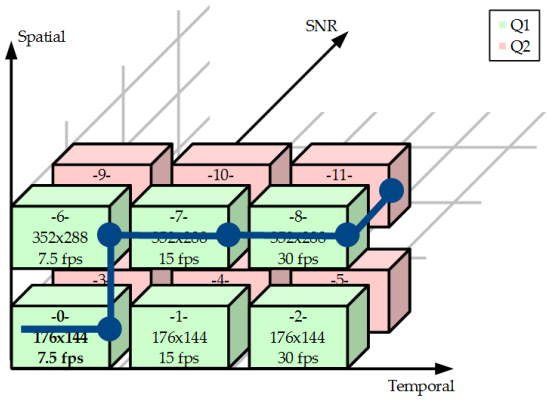


Figure 4.11: SVC Operating Points

4.3.1.5 *Video Encoding & Layers Extraction*

Videos are SVC-encoded using the JSVM official reference software implementation [2]. JSVM outputs a file or stream of NAL Units (NALUs), which are combined in subsets to form individual SVC layers [54]. To prepare data for the dissemination scheme, the different layers have to be extracted out of the stream. For this, JSVM provides a bit-stream extractor function that extracts a specific layer. But undesirably, the returned stream does not include the specified layer only, but also all of the dependent successive layers. Hence, in order to extract physically independent layers with no redundancy with each other, a modification to JSVM's implementation of the extraction function has to be applied to ignore dependencies and extract a single layer at once. This change is presented by Chen et al. [117]. Consequently, any layer (shown in Figure 4.9) or subset of layers (OPs shown in Figure 4.11) are extractable as a separate file, which can be individually sent.
By applying the aforementioned SVC configurations on the sample video sequences: V_0, V_1 and V_2, the resulting sizes and PSNR quality values of the given OPs are as listed in Table 4.1.

OP	*PSNR (db)*			*Size (KB)*		
	V_1	V_2	V_3	V_1	V_2	V_3
1 (Base)	25.3	28.7	28.7	86	118	236
2	26	30.2	30.2	263	685	1370
3	29	32.3	32.3	44	110	220
4	33	34.9	34.9	91	186	372
5	36	37.3	37.3	735	2321	4642

Table 4.1: Sizes and Quality Values of the Operating Points

The PSNR values are not calculated for SVC in a straight forward way compared to unscalable videos as given in Section 2.3.3. For multistream videos, PSNR has to be calculated for each of the substreams (layers) separately. The problem is that each

substream has a specific value for the $Width_{frame}$, $Height_{frame}$ and N_{frames} parameters, depending on its spatial and temporal resolution. The reference video has the upper bound values for those parameters. A substream of a lower spatial layer will have smaller *W* and H values, whereas a substream of a lower temporal layer will have a smaller N value. Consequently, lower layers have to be upsampled temporally and/or spatially to match the reference video and hence be applicable in Equation 2.3 [118]. This technique is described in details in Appendix A.2.

4.3.2 *Simulation Environment*

4.3.2.1 *Mobility*

As presented in Section 2.2.3, to avoid the drawbacks of using real-world traces, we use the Shortest Path Map-Based Movement (SPMBM) model in a map-based area. The map is taken from Helsinki city, which is the default built-in map for the Opportunistic Network Environment (ONE) simulator with predefined streets and POIs. The size of the map is 4300 × 3400 meters including the free space that occupies most of the map as shown in Figure 4.12.

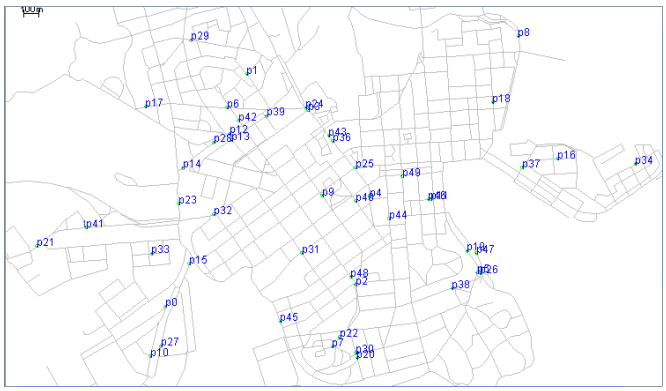
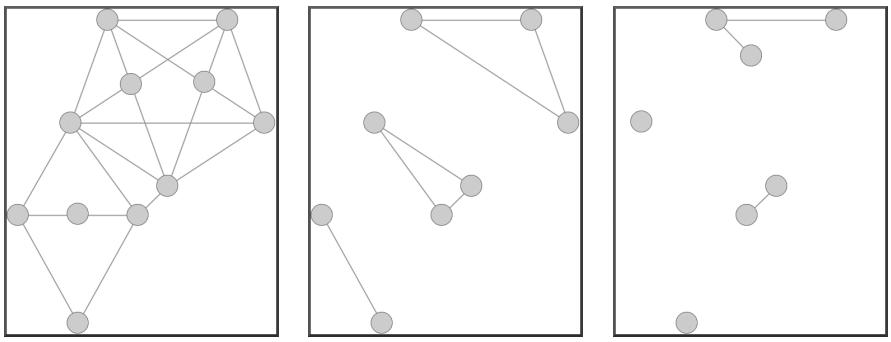


Figure 4.12: Helsinki Map

Furthermore, to keep the common comparison reference, we also run our experiments using the Random Way-Point (RWP) mobility model in an open area of the size 1000 × 1000 meters.
Besides, the simulation run (and mobility) lasts for 360 minutes. Although it is more realistic to be limited to 180 or 240 minutes, we implement this long run in order to examine the quality impact of the late arriving ELs.

4.3.2.2 *Population*

As introduced in Section 2.1.1, the density of wireless networks is classified into three levels: dense, medium and sparse, which are visually illustrated in Figure 2.1.
OppNets fall in the third category of sparse networks, where nodes are scattered with



(a) Dense (b) Medium (c) Sparse

Figure 4.13: Density Patterns of Wireless Networks

only a few number of active connections at any instant of time. However, in the simulated experiments in the literature, the used number of nodes varies from 50 to 1000 or more nodes [119, 22]. High populations are criticized by Grasic and Lindgren [21] due to the unrealistic usage of big number of nodes. Consequently, to keep the sparse settings of OppNets, we set our population to 50 pedestrian nodes.

4.3.2.3 *Routing*

As presented earlier in Section 4.2.3, the SnW routing protocol (in its both variations: Vanilla and Binary) is chosen because it is considered as a reference routing schema and it can easily implement the intended unequal dissemination.
Furthermore, to keep a baseline and a reference for comparisons, the epidemic and PRoPHET protocols[1] are also tested. Epidemic, as an uncontrolled flooding protocol, delivers the best results in a given scenario with regard to delay and delivery ratio when the messages buffer is big enough. PRoPHET, on the other hand, is an estimation-based protocol, and hence gives the chance to test another family of routing protocols.
All the SnW, Epidemic and PRoPHET routing protocols are built-in delivered with the ONE simulator, and can be directly used with no extra costs of further implementations. However, the basic implementation of SnW allows to set only one redundancy factor for all the generated messages. Therefore, in order to set different redundancy factors according to the importance degree of each message, the open-source simulator had to be adapted as described in Appendix A.3.2.

4.3.2.4 *Other Configuration Parameters*

The transmission interface is set to Bluetooth with a 250kBps transmission rate, which represents a lower-bound that can be ensured to exist on any mobile device.

1 Section 2.1.2.

Messages' Time-to-Live (TTL) is set unlimited (or in other words, limited to the simulation lifetime), because we are interested in examining late arriving messages, and because messages in our scenarios have no validation deadline.
Lastly, the buffer size on each node is set to 100 MB, so that it is big enough to carry all forwarded messages.

4.3.2.5 *Summary*

All experiments are performed on the Opportunistic Network Environment (ONE) simulator [1]. A summary of the configuration parameters is given in Table 4.2.

Parameter	Value	Remarks
Mobility	Random Way-Point (RWP)	Open area: 1000×1000m
	Shortest Path Map-Based Movement (SPMBM)	Helsinki map with predefined streets and POIs
Run duration	360 min	-
Population	50 pedestrian nodes	Ensure a sparse setup
Routing	Spray-and-Wait (SnW)	Vanilla and Binary
	Epidemic	Baseline for comparisons
	PRoPHET	Further comparisons
Transmission interface	Bluetooth	Rate: 250kBps, Range: 10m
TTL	= Run duration = 360 min	-
Buffer	100 MB	-

Table 4.2: Simulation Configuration

Lastly, to reduce variability, every experiment is repeated in $N_r = 400$ complete simulation runs (rounds), each time with different randomly selected source and destination nodes. N_r is statistically set, so that the standard deviations of the delivery times are not significantly affected by a higher value of N_r.

4.4 MAP-BASED EXPERIMENTS

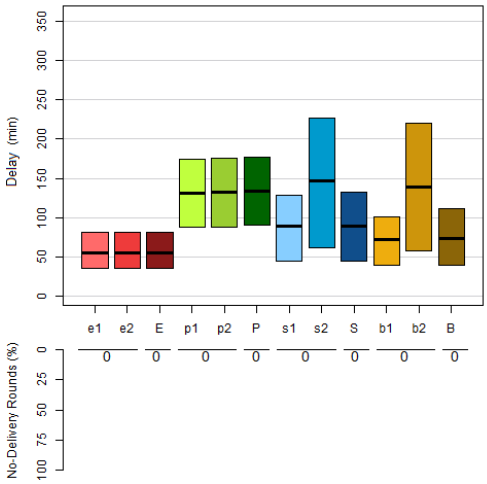
This section presents and analyzes the results of the map-based experiments. The selected video sequences are disseminated first as one unlayered part, and then SVC-encoded using the proposed scheme.

4.4.1 *Delivery Ratio and Delay*

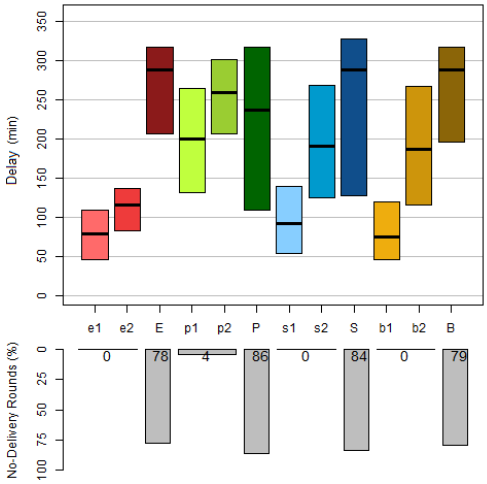
Results for the three video sequences are depicted in Figure 4.14. Box-plots that refer to the 25^{th}, 50^{th} and 75^{th} percentiles are used, along with bars referring to the percentage number of rounds where no delivery was reported.

In all Figures, the different routing protocols are represented by different color sets. The depicted boxes are denoted as follows:

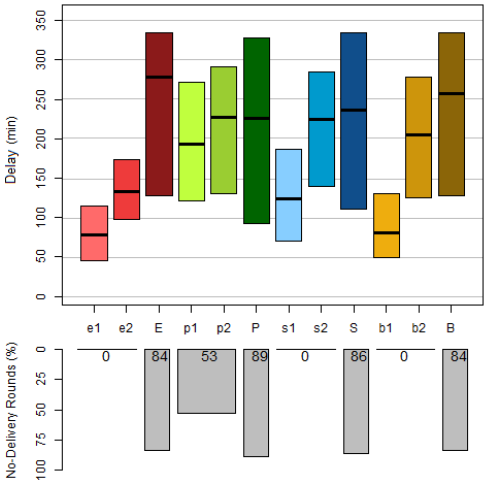
- e1, e2, E: SVC's base layer, highest delivered enhancement layer and the non-SVC video, in respective order, using Epidemic routing
- p1, p2, P: same as before using PRoPHET routing
- s1, s2, S: same as before using vanilla SnW routing
- b1, b2, B: same as before using binary SnW routing



(a) V1



(b) V2



(c) V3

Figure 4.14: Map-Based Experiments

The consequence of the proposal can be concluded out of the box-plots for the larger videos V2 and V3, by comparing the x_1 and x_2 SVC segments to non-SVC X segment,

where a triplet $\{x_1, x_2, X\}$ corresponds to one routing algorithm (e.g. $e1, e2, E$). For example, using *binary* SnW, b_1 represents the time that the destination node has to wait before it can start watching a video at its lowest quality:

- $Avg_{v2}(b1) \approx 70$ min, on Figure 4.14b
- $Avg_{v3}(b1) \approx 80$ min, on Figure 4.14c

Video quality keeps getting better until b_2, the delivery time of the last successfully delivered enhancement layer.

- $Avg_{v2}(b2) \approx 190$ min, on Figure 4.14b
- $Avg_{v3}(b2) \approx 210$ min, on Figure 4.14c

On the other hand, B is the delay for delivering the non-SVC version.

- $Avg_{v2}(B) \approx 280$ min, on Figure 4.14b
- $Avg_{v3}(B) \approx 260$ min, on Figure 4.14c

It is noticed that in general $x_1 \ll X$ and $x_2 < X$. Moreover, the no-delivery ratio, represented below each figure, is much higher for non-SVC transmission.
Comparing the different routing algorithms, we can note that results of *binary* SnW (following our approach) are equal to or slightly less good than the results of *Epidemic*, although the network overhead for the *binary* SnW case is much lower than *Epidemic*'s overhead. The PRoPHET protocol, on the other hand, is the worst among others with regard to both delay and delivery. This shows that redundancy-based routing protocols suits better for sparse environments, because PRoPHET needs more contacts in order to enhance its estimation function.
The percentile margins of the box plots around the corresponding averages are noticed to be generally wide referring to dispersed data. This common problem is due to characteristics of OppNets, where the delivery probability and delay vary a lot. Increasing the number of simulation rounds does not completely help to overcome this phenomena.
Besides, the playout availability, which reflects the availability of any playable version of the video at any quality, is equal to 100% in both cases (0% no delivery rounds). This is achieved in our scheme with the aid of the focus on delivering the BL, which ensures playing the video at the lower quality.
The main advantages of our proposal with regard to the given example scenarios are:

1. Less playout delay: so that an informing video (e.g. about a demonstration or a disaster) can be delivered at a lower quality but with the least possible delay.
2. High playout availability: so that the probability to receive the informing video at least is high.

3. QoE taken into consideration: by the means of video reception on 2 levels (lower quality at lower delay and higher quality at higher delay).

4.4.2 *Quality Distribution*

One drawback when using SVC is that the best successfully delivered quality is not necessarily always the maximum quality that is achieved when delivering all of the enhancement layers. In some simulation rounds, only the base layer could be delivered, so the quality stops at the base level. Other rounds reached the quality level associated with further OPs. For instance, Figure 4.15 presents the quality distribution for the three video sequences when using *Epidemic* and *binary SnW* routing algorithms.

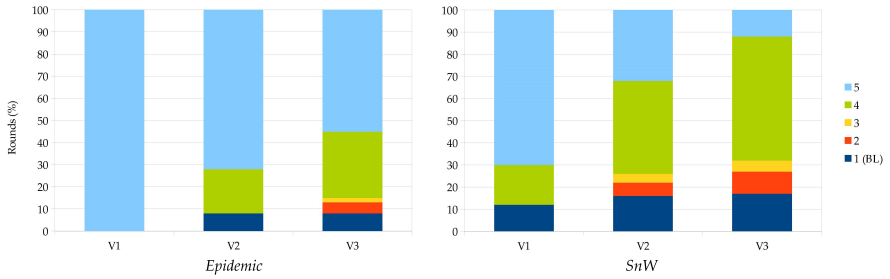


Figure 4.15: Quality Distribution

The video V3 could reach the maximum quality (the 5^{th} OP) in 55% of the simulation rounds using *Epidemic*, where 30% stopped at the quality level of the 4^{th} OP and less than 10% at the BL quality. On the other hand, using *binary SnW*, only 12% of the simulation rounds reached the 5^{th} OP, where over 50% stopped at the quality level of the 4^{th} OP, less than 10% at each of the 3^{rd} and 2^{nd} OPs, and about 18% at the BL quality.
The same analysis above applies for each of the other videos V1 and V2. Consequently, average quality levels are calculated and listed in Table 4.3 with regard to the reached OPs and the corresponding PSNR values.

Metric	*Epidemic*			*Binary SnW*		
	V_1	V_2	V_3	V_1	V_2	V_3
OPs	5	4.5	4.2	4.3	3.7	3.4
PSNR	36	36.1	35.4	34.2	34.3	33.5

Table 4.3: Average Quality Levels

4.4.3 *Viewing QoE*

4.4.3.1 *Results Normalization*

The viewing experience is evaluated using the Equation 4.1 from Section 4.2.2, after the measurement values are normalized using the Equation 4.2.

QUALITY The achieved average PSNR values from Table 4.3 are normalized on the basis of the videos' full PSNR values in Table 4.1 as the range upper bound (lower bound = 0). To invert the quality values, so that the optimal value lies at zero (as described in Section 4.2.2), the Equation 4.3 is applied.

$$\tilde{q} = -1 \times \tilde{PSNR} + bound_{upper} = -1 \times \tilde{PSNR} + 1 \tag{4.3}$$

Consequently, the values of the normalized PSNR and the normalized inverted quality are given in Table 4.4.

Metric	*Epidemic*			*Binary SnW*		
	V_1	V_2	V_3	V_1	V_2	V_3
$\tilde{PSNR}$	1	0.97	0.95	0.95	0.92	0.90
$\tilde{q}$	0	0.03	0.05	0.05	0.08	0.1

Table 4.4: Normalized Average Quality Levels

DELAY The delay values are normalized on the basis of the messages' set TTL as the upper bound. The lower bound is set to 0.

OVERHEAD The overhead of an experiment is calculated as the following:

$$overhead = \frac{I_{data}}{D_{data}} \times \frac{1}{overhead_e}$$

where,
D_{data} : The amount of successfully delivered data
I_{data} : The initiated amount of data, including the redundancy of the "controlled-flooding" routing
$overhead_e$: The overhead of the Epidemic routing protocol
Then, all overhead values are normalized on the basis of the overhead of the Epidemic routing protocol, which represents the upper bound of the possible overhead in the network. Similarly, the lower bound is set to 0.

4.4.3.2 *Calculation Of The Viewing QoE*

As proposed, two measurement points (p_1, p_2) are taken, referring to 1) the result associated with the lowest playout delay at the lowest quality, and 2) the result associated with the maximum delivered quality at any delay. Consequently, an analytical summary of the viewing QoE values is given in Table 4.5.

		Epidemic			*Binary SnW*		
		V_1	V_2	V_3	V_1	V_2	V_3
p_1	PSNR	25.3	28.7	28.7	25.3	28.7	28.7
	PSÑR	0.70	0.77	0.77	0.70	0.77	0.77
	$\tilde{q}$	0.3	0.23	0.23	0.3	0.23	0.23
	t (min)	55	80	80	74	80	80
	$\tilde{t}$	0.153	0.222	0.222	0.206	0.222	0.222
	$\tilde{o}$	1	1	1	0.15	0.15	0.15
	$val(p_1)$	1.055	1.049	1.049	0.393	0.353	0.353
p_2	PSNR	36	36.1	35.4	34.2	34.3	33.5
	PSÑR	1	0.97	0.95	0.95	0.92	0.90
	$\tilde{q}$	0	0.03	0.05	0.05	0.08	0.1
	t (min)	55	115	130	145	190	210
	$\tilde{t}$	0.153	0.319	0.361	0.403	0.528	0.583
	$\tilde{o}$	1	1	1	0.15	0.15	0.15
	$val(p_2)$	1.012	1.050	1.064	0.433	0.554	0.611
QoE(p)		1.033	1.050	1.057	0.413	0.454	0.482

Table 4.5: Viewing Experience Evaluation

As Figure 4.2 shows, the lower the value of the viewing QoE, the better it is. It is noticed using this metric how each of the parameters (quality, delay and overhead) contribute to the QoE. Consequently, using the *Binary SnW* routing protocol along with our proposal that is summarized in Listing 4.1, the results in Table 4.5 are significantly better than those of *Epidemic*, although this latter has slightly lower delays and better qualities, but a high overhead.

Beside the metric of the viewing QoE, our approach of using a layered video coding and prioritizing the more important parts enables the scheme to deliver a quality level that is very close to the quality level of *Epidemic*, which represents the performance upper bound since the message buffer is assumed to be big enough. This is noticed in Table 4.5 for p_2 under the results of direct and normalized PSNR.

Furthermore, to overcome the high delay to reach high levels of quality (for instance 210 min for V_3 using SnW compared to 130 min using *Epidemic*), the layered video encoding enables the scheme to deliver a degraded and smaller version (that is represented by p_1) at significantly lower delays, which positively contributes to the viewing QoE.

4.5 OPEN-AREA EXPERIMENTS

This section presents and analyzes the results of the open-area experiments. The selected video sequences are disseminated first as one unlayered part, and then SVC-encoded using the proposed scheme.

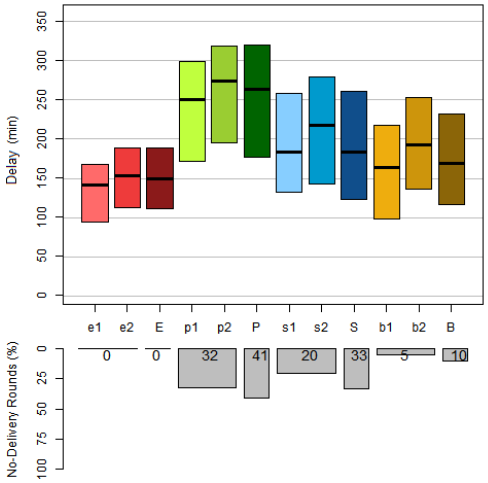
4.5.1 *Delivery Ratio and Delay*

Results for the three video sequences using the RWP mobility are depicted in Figure 4.16. Box-plots that refer to the 25^{th}, 50^{th} and 75^{th} percentiles are used, along with bars referring to the percentage number of rounds where no delivery is reported. The interpretation of the figures is the same as given in Section 4.4.1.
The results for the applied open-area settings are worse than those for the map-based settings in the previous section. The reason behind this is the low number of contacts because of the (completely) random movement. The most remarkable setback is the delivery ratio, which leads to imprecise comparisons of the delay results since they are only calculated for the successfully delivered data. For instance, for the V_3 video, the delay of e_2 (max quality using Epidemic) is higher than the delay of s_2 (max quality using binary SnW). This is not realistic, but it is because that the results for Epidemic are calculated among 96% of the simulation rounds (4% no-delivery rounds), whereas the results for the binary SnW are only calculate among 80% of the simulation rounds.
However, similar conclusions to those in Section 4.4.1 can drawn, with regard to:

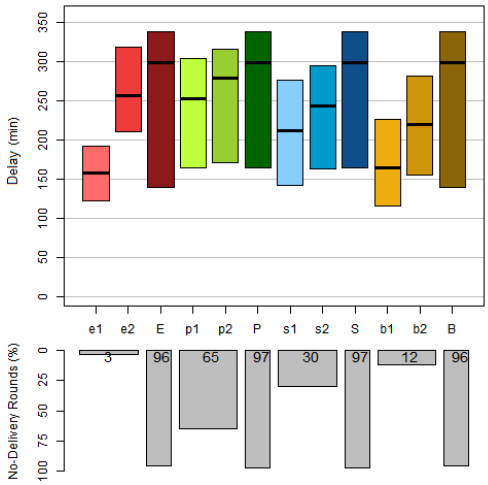
- Comparing the x_1 and x_2 SVC segments to the non-SVC X segment.
- The superiority of the binary SnW routing protocol among other protocols, contributing to:
  - Less playout delay.
  - Higher playout availability.

4.5.2 *Quality Distribution*

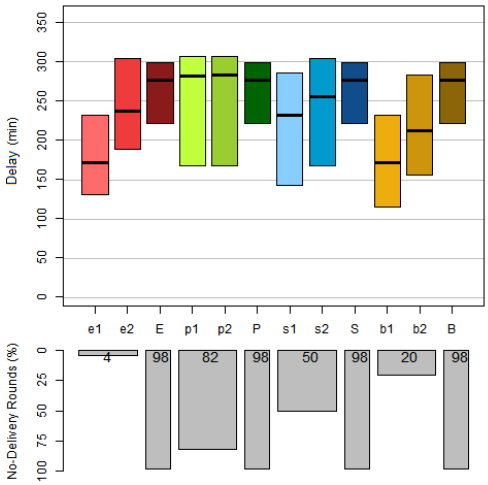
The average quality levels are calculated and listed in Table 4.6 with regard to the OPs and the corresponding PSNR values.



(a) V1



(b) V2



(c) V3

Figure 4.16: Open-Area Experiments

Metric	*Epidemic*			*Binary SnW*		
	V_1	V_2	V_3	V_1	V_2	V_3
OPs	5	4.1	3.6	4	3.1	2.8
PSNR	36	35.1	33.7	33	32.5	31.8

Table 4.6: Average Quality Levels (Open-Area)

4.5.3 *Viewing QoE*

4.5.3.1 *Results Normalization*

The viewing experience is evaluated using the Equation 4.1 from Section 4.2.2. The measurement values for quality, delay and overhead are normalized in same way as

in Section 4.4.3.1. The values of the normalized PSNR and the normalized inverted quality are given in Table 4.7.

Metric	*Epidemic*			*Binary SnW*		
	V_1	V_2	V_3	V_1	V_2	V_3
PSÑR	1	0.94	0.90	0.92	0.87	0.85
$\tilde{q}$	0	0.06	0.1	0.08	0.13	0.15

Table 4.7: Average Quality Levels (Open-Area)

4.5.3.2 *Calculation Of The Viewing QoE*

As proposed, two measurement points (p_1, p_2) are taken, referring to 1) the result associated with the lowest playout delay (corresponding to the lowest quality), and 2) the result associated with the maximum delivered quality. An analytical summary of the viewing QoE values is given in Table 4.8.

		Epidemic			*Binary SnW*		
		V_1	V_2	V_3	V_1	V_2	V_3
p_1	PSNR	25.3	28.7	28.7	25.3	28.7	28.7
	PSÑR	0.70	0.77	0.77	0.70	0.77	0.77
	$\tilde{q}$	0.3	0.23	0.23	0.3	0.23	0.23
	t (min)	145	155	170	160	160	170
	$\tilde{t}$	0.403	0.431	0.472	0.444	0.444	0.472
	$\tilde{o}$	1	1	1	0.15	0.15	0.15
	$val(p_1)$	1.119	1.113	1.129	0.557	0.522	0.546
p_2	PSNR	36	35.1	33.7	33	32.5	31.8
	PSÑR	1	0.94	0.90	0.92	0.87	0.85
	$\tilde{q}$	0	0.06	0.1	0.08	0.13	0.15
	t (min)	155	255	240	190	220	215
	$\tilde{t}$	0.431	0.708	0.667	0.528	0.611	0.597
	$\tilde{o}$	1	1	1	0.15	0.15	0.15
	$val(p_2)$	1.088	1.227	1.206	0.554	0.642	0.634
QoE(p)		1.104	1.169	1.167	0.555	0.582	0.590

Table 4.8: Viewing Experience Evaluation (Open-Area)

Conclusions from Figure 4.16 and Table 4.8 do not differ from those from Figure 4.14 and Table 4.5. Most importantly is how using our proposed metric allows each of the quality, delay and overhead parameters to contribute to the QoE. Moreover, how the layered video encoding enables the scheme to deliver a degraded and smaller version at lower delays, which positively contributes to the viewing QoE.

4.6 VALIDATION AND COMPARISON

This section aims to validate our results by comparing them to some of the corresponding works from Section 3.2. The target works are those that are implemented in DTNs or OppNets. Nevertheless, no direct numerical comparisons can be drawn because of the many different setting parameters. Instead, we try to estimate the results of the presented works as if they were performed in our settings.
On the other side, for all other delivery schemes in MANETs it is not possible to draw a comparison or a conclusion analysis because of the different settings that are meant for the stable connectivity of MANETs.

Raffelsberger and Hellwagner [81] use a very expensive encoding technique (MPEG-based[2]) that must be repeated for each desired quality level. They limit the quality levels to only 2, where the second level is playable by itself independently of the first one. This flexibility comes at the expense of a large size for the higher level compared to SVC. For instance, if a video is encoded using SVC into 2 layers with the sizes s_1, s_2 respectively, the whole video will have the total size of $s_1 + s_2$, whereas using the technique in [81] the video will have the size of $s_1 + (s_1 + s_2)$ because of the redundancy in the second level. Hence, the overhead can grow dramatically with the increasing number of layers (e.g. 5 in our experiments). This is depicted in Figure 4.17, which estimates the sizes of the 5 OPs of the video V_1 if the technique in [81] was applied, compared to the sizes of our OPs from Table 4.1.

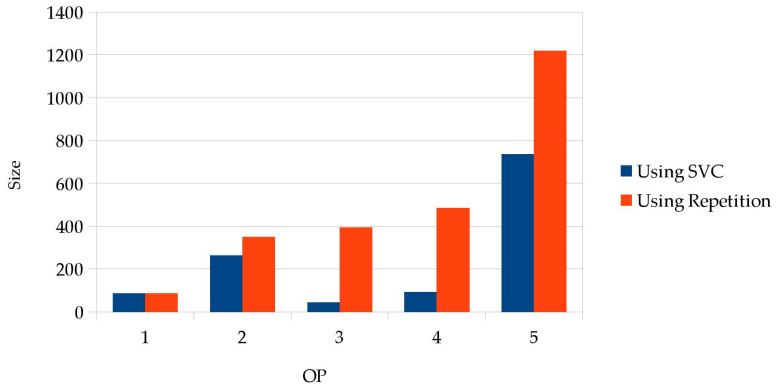


Figure 4.17: Overhead Comparison

It is easily noticed how using SVC leads to a lower overhead. Besides, to compare the influence of the authors' work on the quality and delay to our work, we build the comparison with regard to the two result points in Figure 4.3.

p_1: The time to deliver the first layer and the resulting quality will not differ from our experimental results that are associated with the delivery of the BL.

2 Not particularly specified.

p_2: For the sake of fragmentation, the authors divide each level temporally into short chunks that can be encoded, decoded and played independently. Hence, the viewing QoE metric for the second result point p_2 can not be successfully applied because both delay and quality are unpredictable. The former because of the jitter between the randomly received playable parts, which makes the playout delay unclearly defined, i.e. there is no particular moment to start watching the whole video. And the latter because of the incapability of PSNR under the absence of a lot of frames as Equation 2.3 shows, that PSNR is calculated among each frame of the video.
Another possibility is to consider that a quality level is successfully delivered only if all of its parts are delivered (while no error concealment is applied). Consequently, the authors' second quality level will have a bad delivery ratio compared to our higher levels because of the larger size due to the encoding technique. Hence, p_2 (and the total viewing experience) is worse than its counterpart in our experiments.

Wu and Ma [84] split single-stream videos in a frame-wise fashion into a number of sub-messages for each frame. They offer a single level of quality, which makes p_1 and p_2 from our viewing QoE metric absolutely identical. In their work, the authors are only interested in the delivery ratio, and no result with regard to delay are reported. However, the offered single level of quality refers to the video's full-quality, which will shift the playout delay to a larger value, which corresponds to the delay of p_2 in our experiments.
The delivered quality is either zero, if a delivery failure occurs, or it is equal to the input encoded quality, if the delivery succeeds. The absence of lower levels of quality is not suitable for environments like OppNets, where loss is common. This is also not suitable for our example scenarios, where delivering a video with a degraded quality would serve for a better viewing QoE.
Lastly, the authors route the sub-messages using the Epidemic protocol, and they control flooding by limiting the buffer size and applying a buffer management technique. Therefore, the network overhead because of further copying the messages can grow dramatically (depending on the used buffer size) up to the overhead levels of using Epidemic in our experiments.
All in all, following the authors' technique in our context will result in a viewing QoE that has:

- a worse delay, to deliver a single high-quality level,
- a better quality that refers to the full video quality, but however with a worse delivery probability,
- and varying overhead, depending on the chosen buffer size.

Chen et al. [82, 83] use Multiple Description Coding (MDC) to encode videos into any number of quality levels. As described in Section 2.3, MDC can achieve a better maximum quality (corresponding to p_2) and a better associated delay since each delivered message can help for the reconstruction. However, MDC makes no differentiation between layers, and hence no special support can be applied for the basic quality. Therefore, by exceptionally protecting the SVC's BL as Chiang et al. [51] recommend, we could achieve a better playout delay (corresponding to p_1), beside a close quality to MDC's basic quality. Besides, the overhead is hard to estimate in comparison to ours, because of combining channel coding with MDC in a controlled-flooding routing protocol.
Consequently, it is not possible to derive a numerical viewing QoE comparison between the authors' and our results. However, looking at the viewing QoE's two result points p_1 and p_2, our approach achieves a better result with regard to p_1, whereas the MDC approach is better with regard to p_2.

4.7 SUMMARY

This chapter addressed the problem of granulation and proposed the Scalable Video Coding (SVC) to encode a video into a number of unequally important layers. These layers are routed then with redundancy factors corresponding to their importance.
An evaluation metric for the viewing experience was defined in a 3D feature space that consists of delivery delay, video quality and network overhead. It was calculated as an average between two feature vectors: the first one refers to the delivery of the BL quality level at the lowest delay, and the second one refers to the delivery of a higher quality level at a higher delay.
Furthermore, the experimental setup parameters were fixed, including: SVC's configurations set of scalability dimensions and levels in each dimension, and the simulator configurations set (population, mobility, routing, transmission, etc.).
Lastly, experiments were performed and evaluated using three different video sequences and two mobility patterns: RWP and SPMBM. Promising results were delivered with regard to the viewing QoE metric. With a very low overhead compared to *Epidemic*'s overhead, our proposed approach using *Binary SnW* could reach a very good playout delay (Figure 4.14). Although the delay for the maximum quality is high, the overall evaluated viewing QoE shows the applicability of the proposed scheme in OppNets under the given specificities (Table 4.5).
Nevertheless, there is still a space for improving the results to overcome two recognizable drawbacks. First, as shown in Figure 4.14, the high delay to deliver the BL as well as the higher quality layers. Second, as shown in Figure 4.15, the limited quality, so that not all the OPs could have always been successfully delivered.

* * *

Chapter 5

ADAPTING GRANULARITY FOR A BETTER CONTACT-OPPORTUNITY EXPLOITATION

5.1 PROBLEM STATEMENT

This chapter addresses the second research problem, which aims to vary and tune the sizes of the video data chunks in a dynamic and adaptive way according to the Opportunistic Networks (OppNets) changing conditions. Using Scalable Video Coding (SVC), as presented in Chapter 4, results in splitting videos into smaller layers, and hence in better delivery results. However, these layers (and the overlaid Operating Points (OPs)) possess predetermined arbitrary sizes according to the encoding configurations. On the other hand, contacts in OppNets are opportunistic, and their durations vary a lot. Therefore, a video layer might be exposed to one of two possible deficiencies with regard its size and the available forwarding opportunities:

- The layer is either too big to be completely sent during a contact between two nodes, then a delivery failure is reported.
- Or it is too small compared to the contact duration, then the opportunity of data transfer is poorly exploited if no further parts are pushed in an aggressive data forwarding way [97]. However, pushing a further layer aggressively, puts this layer again against the same two possible deficiencies.

In this context, we argue that a further granulation of video layers into smaller parts can partially resolve the problem in both cases. First, smaller data parts can better fit into short contact durations and hence avoid the first deficiency with a high probability. Second, aggressive data forwarding can be supported more easily, since a number of small data parts can most likely be pushed before a disconnection occurs. However, applying a high degree of granulation introduces a new drawback of subjecting the whole video to a higher partial loss probability, which can result in delivery failures and worst-case delays.

Hence, the size of the data chunk serving as a transmission unit should be determined adaptively, depending on the changing network conditions of the OppNet. However, the determination of the network conditions in such a fully decentralized environment is quite difficult, since there is no central controller and nodes interact only with their changing neighbors. Consequently, this chapter introduces an adaptive solution, which enables each node in the network to predict and prepare the next optimal chunk size as precisely as possible, based on a dynamic monitoring of environmental parameters.

5.2 OBJECTIVES AND CONTRIBUTIONS

This section highlights the objectives and contributions of this chapter. First, it discusses an alternative approach for granulation, which goes beyond scalability layers. Then, the need for adaptivity is introduced, and propositions are discussed.

5.2.1 *More Efficient Granulation*

Experiments of Chapter 4 were concluded with two main drawbacks: a limited quality level and a high delay. One open issue is how to determine the granularity of data transmission. Indeed, while SVC layers (and the overlaying OPs) provide a good abstraction of the data related to different levels of video quality, some OPs can still be relatively large in size (up to 40-50% of the whole video, as shown in Table 4.1). Hence, due to the unreliability of the opportunistic networks, trying to transmit such an OP at once has a higher probability of failure, which limits the performance of our approach. Whereas smaller transmission units would be more appropriate to overcome the problem of short contact times.
Consequently, we propose to reduce the granularity of transmission units (on the application-layer) down to NAL Units (NALUs) that compose the SVC layers. The advantage of choosing NALUs in this context is that there is no more cost for the partitioning process, because NALUs are the atomic elements of the SVC stream by default [54]. Moreover, the concepts of unequally important parts and prioritized routing can still be applied for NALUs in the same way as before in Chapter 4 for SVC layers. Each NALU represents one frame at one specific layer, and hence can inherit the corresponding importance degree, which will affect later the redundancy factor of the routing protocol.

5.2.1.1 *The Number and Distribution of NALUs in a Video Stream*

The number of NALUs in an SVC stream is determined by the stream's number of frames and the given scalability options (dimensions and levels). In general, there

will be a big number of NALUs, over which the total video size will be distributed. For example, using the fixed configurations in Chapter 4, Figure 5.1 depicts the distribution of NALUs in one Group-of-Pictures (GoP) of 4 frames.

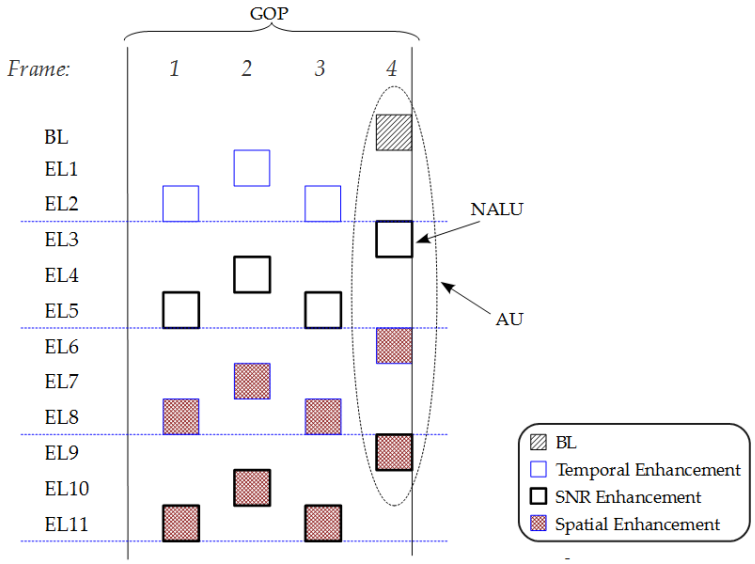


Figure 5.1: NALUs in One GoP of an SVC Video

It can be noticed that for 4 frames within one GoP, there are 16 NALUs among the 12 SVC layers. That is, the Base Layer (BL) has 1 NALU (and hence 1 frame) per GoP. To enhance the temporal resolution, the Enhancement Layers EL-1 and EL-2 double the number of frames in the GoP by adding one then two NALUs respectively. EL-3, -4 and -5 enhance their previous counterpart NALUs on the Signal-to-Noise Ratio (SNR) scalability dimension. Similarly, the ELs -6 to -11 build on the same structure to finally result in 16 NALUs representing 1 GoP of 4 frames. The same applies for all next GoPs of the video. Thus, for the whole video there would be NALUs equal to 4 times the number of frames.

Example 5.1. A video with 100 frames, 12 SVC layers and GoP=4, will have 25 GoPs, each with 16 NALUs. Hence, the BL will have one NALU per GoP $= 25 \times 1 = 25$ NALUs, and the whole video will have $25 \times 16 = 400$ NALUs, over which the size of the video will be divided.

Consequently, our used sample videos will have the given distribution of NALUs in Table 5.1.

Video	# Frames	# Layers	GoP	# NALUs in BL	# NALUs in Vid.
V_1	600	12	4	150	2400
V_2	2000	12	4	500	8000
V_3	4000	12	4	1000	16000

Table 5.1: The Distribution of NALUs in The Sample Videos

5.2.1.2 *The Transmission of NALUs*

The size of a single NALU varies a little bit from one to another, but they are all of very small sizes compared to the sizes of SVC layers. Example 5.1 shows how the size of the video can be distributed over 400 NALUs instead of only 12 layers.
Consequently, an aggressive data forwarding becomes easy to implement: when coming in contact with another node, say x, send one messages (a copy of an NALU). If the contact with x is still ongoing, send a further message. This loop continues until the contact is broken. The small sizes of NALUs will allow to better exploit the contacts and send more units before disconnection, without an extra partitioning overhead to generate the already existing NALUs. Nevertheless, the big number of NALUs, each of which serving as a single transmission unit, results in a discouraging high degree of granulation for the use-cases where a partial loss would dramatically affect the total delivery [88, 120]. For instance, when sending an SVC layer, if any NALU of this layer is lost (while no loss concealment is assumed to be applied[1]), other received units are useless.
To express the problem of a high degree of granulation mathematically, let be:

$$P(Layer_i) = q \quad : \text{the probability of delivering a layer (i) as one part}$$
$$P(NALU_{(i,j)}) = p_j \quad : \text{the probability of delivering a NALU (j) of the layer (i)}$$
$$P(NALU_i) = \prod_{1}^{n} p_j = p \quad : \text{the probability of delivering all the NALUs of a layer (i)}$$

where n is the layer's total number of NALUs, so that for any SVC video: $n \gg 1$, as shown in Table 5.1 and Example 5.1.
Due to the characteristics of OppNets, the delivery probability can not in general reach its upper bound, i.e. $p_j < 1$ and $q < 1$. Furthermore, because of the difference in size between a NALU and an SVC layer: $p_j > q$. Although p refers to delivering the composing NALUs of the layer whose delivery probability is q, we can not assume that $p = q$, because the NALUs can take different paths and thus have independent delivery probabilities. Moreover, for a big enough $n \gg 1$ then $p \ll 1$ and $p < q$.
Therefore, an SVC layer has to be divided into a number of chunks, m, less than the number of NALUs composing this layer: $1 < m < n$. Let $P(chunk_k) = r_k$ be the probability of delivering a newly resulting chunk (k), then we have to find the optimal number of chunks (m) that yields:

$$\prod_{1}^{m} r_k > q > p$$

1 Later in Chapter 6, a concealment mechanism will be studied as a further contribution of this work to overcome partial losses.

5.2.2 *Adaptivity*

Concluded from the last section, the transmission units must lie with respect to their number and size in the range between the few but big SVC layers and the lots of but small NALUs. In other words, if the number of layers composing a scalable video is l, and the number of NALUs is n, we have to find the optimal number of chunks, m, which yields a higher delivery probability than the other cases when sending layers or NALUs, so that $l < m < n$. Tuning the optimal number of chunks takes place by determining the optimal chunk size depending on the network conditions. The advantage of having the NALUs in this context is that the number and size of the transmission units can then be dynamically adjusted by packetizing the very small units into larger chunks. Furthermore, each node in the network can perform the packetization process adaptively according to the environmental parameters on its side, which would reflect the node's knowledge about network conditions like density and mobility.
To develop an adaptive solution, there is a variety of environmental parameters that can play a role in the dynamic characterization in the OppNets. As presented in Section 3.3, these parameters include the contact time (or duration), inter-contact time and contact volume. The inter-contact time can serve for analyzing the frequency of contacts and estimating the occurrence of the next contact. The contact volume studies the stability of contacts. On the other hand, the contact time represents how long a contact between two nodes lasts. By estimating future contact times, the optimal granularity level with regard to the chunks number and size can also be estimated. Since the transmission interface and its transfer rate are fixed, then the sought chunk size can be calculated as in Equation 5.1.

$$size = contactTime \times transferRate \tag{5.1}$$

Besides, the calculation of the environmental parameters can use different settings regarding *1) dynamicity*:

- Semi-dynamic setting: for relatively stable environments, the environment parameters are not set in advance, but they are calculated over a short time then they are fixed and considered to be valid for the future. Big changes in network conditions are not expected.
- Dynamic setting: for continuously changing environments, parameters are regularly updated all over the time.

and *2) the calculation domain*:

- Local: for sparse environments, each node calculates and keeps its own environments parameters. Exchanging these information with other nodes is excluded.

- Distributed: for well-connected environments, the knowledge of other users is taken into account by exchanging their environments parameters upon contacts.

Within the requirements of our proposition, calculations are performed dynamically and locally, which is more reliable in the instable environment of OppNets with rare opportunistic contacts.

5.3 CONTACT-BASED ADAPTIVE PACKETIZATION

This section overviews the proposed adaptive solution. It enables the nodes to predict future contact times, derive the suitable chunk size and then perform packetization on the small NALUs into single transmission units of the corresponding calculated size.

5.3.1 *Contact Time Prediction*

In OppNets, where no fixed topology exists, the network conditions differ from one location to another, and from one node to another. For example, because of the continuous mobility, some parts of the environment might temporarily become denser than other parts, and hence contacts would be more frequent. Moreover, some nodes might be moving faster than other nodes, which makes the contacts last shorter. Therefore, the environmental parameters which reflect those conditions have to be measured at each node separately. Similarly, the calculations and the process of packetization are to be performed by each node independently. Each node of the network keeps track of its own contact time history, and prior to packetization it tries to predict the next contact time by calculating the target value out of the history list up to the most recent past. This predicted target value is meant to serve as a lower limit for the next contact time, so that the actual contact time has a low probability to last shorter than this limit.
According to Gao and Cao [121], the distribution of contact times in Delay Tolerant Networks (DTNs) approximately follows a normal law, therefore each node is proposed to calculate only the contact times' mean (μ) and standard deviation (σ), and keep updating them regularly. Updates take place after every disconnection, to push the last contact into the history list. Based on the up-to-date μ and σ values, the prediction of the next contact time is done then using the 3-sigma rule on the normal distribution [122]. This rule is based on the fact that an observation x lies around the mean value within distances that are defined by the standard deviation with the following probabilities:

$$P(\mu - \sigma \leqslant x \leqslant \mu + \sigma) \approx 0.682$$

$$P(\mu - 2\sigma \leqslant x \leqslant \mu + 2\sigma) \approx 0.954$$

$$P(\mu - 3\sigma \leqslant x \leqslant \mu + 3\sigma) \approx 0.997$$

The different distances are depicted in Figure 5.2, along with corresponding probabilities for x to lie within the given distance.

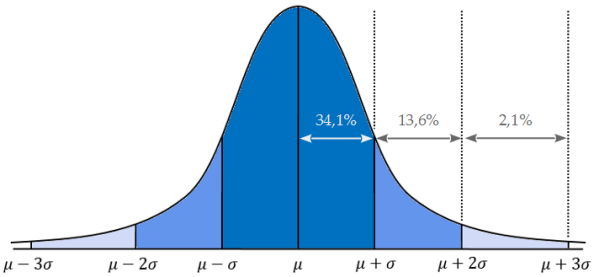


Figure 5.2: The 3-Sigma Rule

However, we are looking for a lower bound value of the contact time, which will high probably be close to or less than the actual contact time to take place next. Therefore, the greater ranges of the previous relations are excluded, leading to the following probabilities:

$$P(\mu - \sigma \leqslant x) \approx 0.841$$

$$P(\mu - 2\sigma \leqslant x) \approx 0.977$$

$$P(\mu - 3\sigma \leqslant x) \approx 0.998$$

A negligible increment of probability is noticed between 2σ and 3σ distances. Therefore, to not over lower the predicted contact time limit, it is fixed as in Equation 5.2.

$$T_{next} = \mu - (2 \times \sigma) \tag{5.2}$$

This relation holds that $P(t \geqslant T_{next}) \approx 97.7\%$, which means that the actual future contact time, t, will most likely (with a certainty of 97.7%) last for no shorter than T_{next}, and thus be enough to transmit the designated chunk size.

5.3.2 *The Process of Packetization*

Once the contact time limit is determined, the corresponding chunk size limit can be calculated from Equation 5.1:

$$size_{limit} = T_{next} \times transferRate$$

and this implicitly defines m, the optimal number of chunks:

$$m = \left\lfloor \frac{size_{total}}{size_{limit}} \right\rfloor \text{ ; } size_{total}\text{: the size if the whole SVC video} \tag{5.3}$$

When the source node has all the NALUs ready, it prepares the initial chunks and transmits them. The number of NALUs in a single chunk is not fixed and varies from one chunk to another, because the NALUs differ in size.
When any further node X, along the way to the destination, receives a chunk of packetized NALUs after a contact with a node Y, it has to record the last contact duration. Then (if necessary), it has to update its μ and σ values, and depacketize the NALUs and repacketize them into one or more chunks of the newly adjusted size limit. That necessity is based on the condition, whether the newly recorded contact duration is much bigger or much smaller than the calculated μ. Otherwise, if the contact duration does not change drastically, the received chunk can remain untouched and ready to be retransmitted. We consider that a new contact duration necessitates an update if it lies outside a distance of σ from the average μ.
The complete algorithm of the above described process is summarized in Listing 5.1.

The contact time (t) with node Z is most likely to be no shorter than T_{next}, and enough to send at least one packet. However, there is still a possibility for t to be:

- $t < T_{next} \Rightarrow$ no chunk will be transmitted
- $t \gg T_{next} \Rightarrow$ more than 1 chunk is transmitted

In all cases, after every contact, the most recent contact time is pushed (if necessary) to the list of the contact times history of both contact's parties, and will influence the newly updated values of their μ and σ parameters. Hence, both nodes will update their T_{next} accordingly. This behavior ensures the dynamicity of the approach when facing changing network conditions, since recent contacts are directly taken into account in the history of the concerned nodes.

5.4 EXPERIMENTS

This section presents and analyzes the results of the conducted experiments. For the continuity of the work, the experimental setup is remained untouched from Section 4.3. However, as no different conclusions were found for the different video sequences, we limit the experiments in this chapter to the V_2: *Highway*, which is fairly long enough to have a big number of NALUs and allow testing our proposal. For the same reason, we limit the mobility model to the *Shortest Path Map-Based Movement (SPMBM)*, and exclude the unrealistic Random Way-Point (RWP). Lastly, the routing protocol is set to *binary Spray-and-Wait (SnW)* only to build on our experimented proposal of prioritized data parts.

```
% When disconnected after receiving a chunk c from Y to X
connection(X,Y).on('disconnection') {
   % Record the last contact duration
   T_last = connection(X,Y).duration;
   if (T_last < (X.μ + X.σ) && T_last > (X.μ − X.σ)) {
      % the contact duration chaged only slightly
      X.buffer_1.push( c ); % buffer1 is for chunks. c is ready to be transmitted
      return; % update nothing.
   }
   % Update the contact time history and corresponding parameters
   X.history.push(T_last);
   X.μ = average(X.history);
   X.σ = stdev(X.history);
   % predict the next contact time, and the corresonding optimal chunk size
   T_next = μ − (2 × σ);
   size_limit = T_next × transferRate
   % depacketize: extract NALUs
   NALUs = depacketize(c);
   X.buffer_2.concatenate(NALUs); % buffer2 is for NALUs
   % do not repacketize now, because more NALUs might be received from other
       nodes than Y
}

% When coming in contact with a new node Z ≠ Y
connection(X,Z).on('set-up') {
   % repacketize the extracted NALUs accordingly
   c = newChunk();
   while (X.buffer_2.notEmpty) {
      NALU = X.buffer_2.pull();
      if (c.size + NALU.size > size_limit) {
         % c reached its size limit, then it is ready to be transmitted
         X.buffer_1.push( c ); % buffer1 is for chunks
         c = newChunk();
      }
      c.add(NALU);
   }
   % there is one last chunk that might not reach size_limit
   X.buffer_1.push( c );
}

% When the connection with Z is up
connection(X,Z).on('connected') {
   while (connection(X,Z).isUp()) {
      c = X.buffer_1.pull();
      connection(X,Z).send(c);
   }
}
```

Listing 5.1: Adaptive Packetiazation at Node X: Reception & Transmission

5.4.1 *Transmission Using NALUs*

As introduced, the drawbacks identified in Chapter 4 lead to the idea of using smaller units than SVC layers. And because an SVC stream is composed by definition of small

NALUs, we performed an experiment to see how feasible it is to use NALUs as the transmission units. Results are given in Figure 5.3.

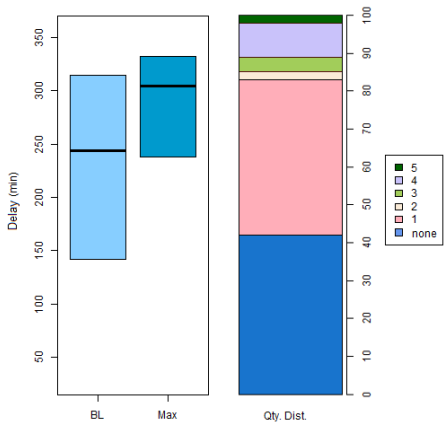


Figure 5.3: Transmission of NAL-Units

The left part of the figure analyzes the delays of delivering the quality of the BL (depicted in light-blue) and the maximum delivered quality level (depicted in dark-blue). Whereas the quality distribution is depicted on the right part of the figure with 5 quality levels referring to the 5 defined Operating Points (OPs). The results are given as a percentage of the simulation rounds that reached the corresponding quality level.
Compared to the results from Chapter 4, the percentage of rounds with no delivery increased from 0% to 42% (i.e. playout availability = 58%), and 34% of the rounds were limited to only the BL quality. This bad result is justified by the very big number of units, which increased the probability of worst-case delay as described in Section 5.2.1.2. For instance, the used video stream has 8000 NALUs under the given configurations. These units are distributed among the different layers (e.g. 500 NALUs for the BL alone). A layer is only playable if all its composing NAL units are successfully received, and hence if any unit of a layer is lost, other received units are useless (with the assumption that no error concealment is applied). Therefore, the more composing parts a video (or a layer) has, the less likely it is to be playable.

5.4.2 *Oracle-Based Experiments*

Before testing our adaptive-granularity approach, an optimal baseline reference is recommended to be built. This optimal case is simulated by running oracle-based experiments. These experiments assume the existence of an oracle, which knows everything about the simulation environment in advance, e.g. contacts occurrences and contacts- and inter-contacts- times. Thats is, given the environmental configurations as the map, the mobility model, the density and the nodes speed, the oracle can

analyze the overall contact times in advance and allow the nodes to estimate their chunks' $size_{limit}$ before launching the simulation. Thus, no calculations have to be performed by the nodes at run-time. Using our given configuration, the distribution of contact times is as shown in Figure 5.4.

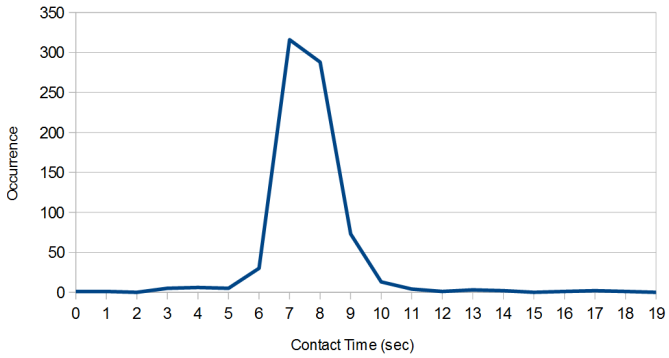


Figure 5.4: Distribution of Contact Times

This distribution holds that $\mu \approx 7.8$ and $\sigma \approx 2$. By applying these values into Equations 5.2 and 5.1, we get [2]:

$$T_{next} = 7.8 - (2 \times 2) \approx 3.8sec$$
$$\Rightarrow size_{limit} = 3.8 \times 250kBps \approx 1MB$$

As an oracle, the chunk size for all nodes during the whole simulation time is fixed to 1 MB, and all nodes are limited to this size when packetizing.
Consequently, by applying our adaptive approach with the above calculated $size_{limit}$ to transmit the NALUs of the video V_2, we got the results in Figure 5.5, including the delivery delays and quality distribution.

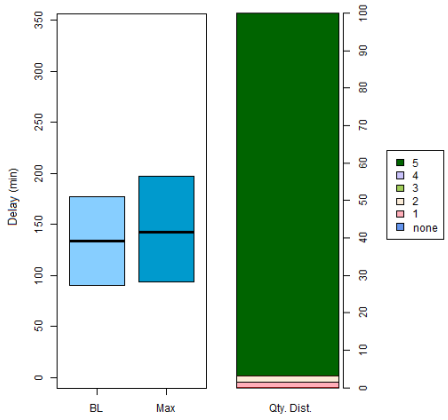


Figure 5.5: Oracle-Based Packetization

The playout delay is the delivery time for BL, and it is worse compared to Figure

2 Transmission interface: Bluetooth, transmission rate= 250 kBps (see Chapter 4)

4.14b from Chapter 4 (75 min vs. 130 min), while the delivery time for the maximum quality is improved (from ≈ 190 in Figure 4.14b to ≈ 145 min). However, the significant improvement lies in the highest achieved quality, which is depicted on the right part of the figure. Over 90% of the rounds reported a full quality (level-5), compared to 34% when using SVC layers (Figure 4.15 from Chapter).
Although having an oracle with a full knowledge is not close to reality, but its experiments serve well for a comparison purpose.

5.4.3 *Adaptive Packetization*

Next, this section applies the adaptive approach on the actual experiments, where each node keeps track of its contact history, and hence also of updated μ and σ values. The next contact time and the corresponding chunk size limit are dynamically predicted as given in Listing 5.1. Simulation are performed with the given configurations, and the results of delay, playout availability, and quality distribution are given in Figure 5.6.

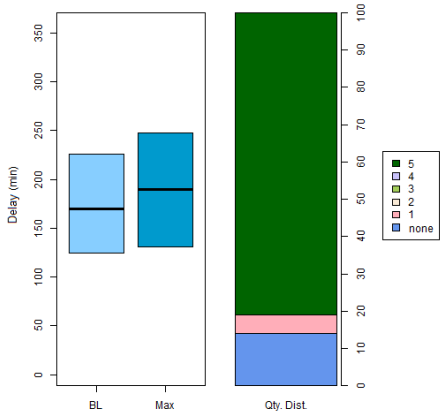


Figure 5.6: Adaptive Packetization

Similarly to oracle-based results, the significant improvement compared to Figure 4.14b appears in the achieved quality level, where 81% of the simulation rounds reach the maximum quality level. On the other hand, the delay of the maximum quality has a slight improvement in comparison to Figure 4.14b, from 190min to ≈ 170min. However, the playout delay reports a setback from 75min to 170min. Another bad result is the 14% loss rate, which lowers the playout availability from 100% to 86%. Both setbacks are related to the BL, whose delayed delivery affects the playout delay, and failed delivery affects the playout availability.
Therefore, to overcome the mentioned setbacks, we propose to combine the two transmission approaches of SVC layers and NALUs adaptive packetization. The idea is to condition the adaptive packetization of the NALUs of a given layer (or OP), so that

it is performed only if the layer (or OP) is larger than the computed threshold size. Hence, if a given OP is already smaller than the computed optimal chunk size, this OP is kept as it is and excluded from extracting its composing NALUs, which are to be fed to the packetization procedure. For the BL, this is particularly important, so that if the BL size does not fulfill the condition (i.e. does not exceed the determined chunk size limit), its NALUs are excluded from being packetized, and the OP is sent at once. Hence, the algorithm in Listing 5.1 is modified accordingly into the one in Listing 5.2.

```
% When disconnected after receiving a chunk c from Y to X
connection(X,Y).on('disconnection') {
   % Record the last contact duration
   T_last = connection(X,Y).duration;
   if (T_last < (X.μ + X.σ) && T_last > (X.μ − X.σ)) {
      % the contact duration chaged only slightly
      X.buffer_1.push( c ); % buffer1 is for chunks. c is ready to be transmitted
      return; % update nothing.
   }
   % Update the contact time history and corresponding parameters
   X.history.push(T_last);
   X.μ = average(X.history);
   X.σ = stdev(X.history);
   % predict the next contact time, and the corresonding optimal chunk size
   T_next = μ − (2 × σ);
   size_limit = T_next × transferRate
   % check the need to depacketize
   if (c.isOP() && c.size < size_limit) {
      % exclude from depacketization
      X.buffer_1.push( c ); % buffer1 is for chunks. c is ready to be transmitted
   }
   else {
      % depacketize: extract NALUs
      NALUs = depacketize(c);
      X.buffer_2.concatenate(NALUs); % buffer2 is for NALUs
   }
}

% When coming in contact with a new node Z ≠ Y
... % untouched from Listing 5.1

% When the connection with Z is up
... % untouched from Listing 5.1
.
```

Listing 5.2: Adaptive Packetiazation at Node X: The Combined Solution

Consequently, by applying the proposed combined solution and running new simulations, the results are refined as shown in Figure 5.7.

The combined solution, compared to Figure 5.6, returns a playout availability of

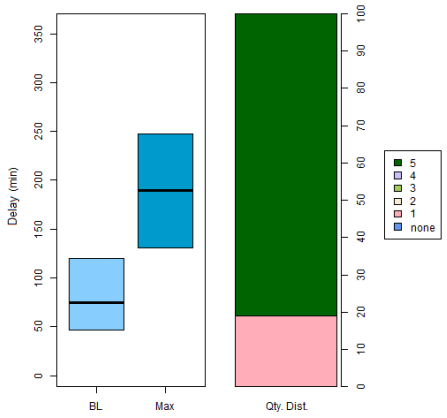


Figure 5.7: Combined Solution: Conditioned Packetization

100%, where 19% of the rounds stop at the BL quality level, and 81% continues to the maximum quality level associated with OP-5. Furthermore, the playout delay becomes equivalent to the one in Figure 4.14b ($\approx$ 75min). Consequently, these results positively contribute for a better viewing Quality-of-Experience (QoE), as presented in the next section.

5.4.4 *Viewing QoE*

5.4.4.1 *Results Normalization*

The viewing experience is evaluated using the Equation 4.1 from Section 4.2.2, after the measurement values are normalized using the Equation 4.2.

QUALITY The average delivered Peak Signal-to-Noise Ratio (PSNR) video quality is given in Table 5.2. Moreover, the same as done in Section 4.4.3.1, the video quality is normalized and inverted, so that the optimal value lies at zero.

Metric	*Binary SnW*
	V_2
PSNR	35.7
PS$\tilde{N}$R	0.96
$\tilde{q}$	0.04

Table 5.2: Average and Normalized Quality Level

DELAY The delay values are normalized on the basis of the messages' set Time-to-Live (TTL) as the upper bound. The lower bound is set to 0.

OVERHEAD One of the important aspects in the proposed adaptive packetization approach is that it does not add extra network overhead onto the previously implemented SVC layers approach. Hence, the calculation and normalization of the overhead remains the same as previously given in Section 4.4.3.1.

5.4.4.2 *Calculation Of The Viewing QoE*

The viewing experience is evaluated as proposed using the Equation 4.1 on two measurement points (p_1, p_2). Analytical summary is given in Table 5.3.

		Binary SnW
		V_2
p_1	PSNR	28.7
	PSÑR	0.77
	$\tilde{q}$	0.23
	t (min)	80
	$\tilde{t}$	0.222
	$\tilde{o}$ %)	0.15
	$val(p_1)$	0.353
p_2	PSNR	35.7
	PSÑR	0.96
	$\tilde{q}$	0.04
	t (min)	180
	$\tilde{t}$	0.5
	$\tilde{o}$ %)	0.15
	$val(p_2)$	0.524
QoE(p)		0.438

Table 5.3: Viewing Experience Evaluation

Consequently, compared to Table 4.5 of Chapter 5, the adaptive solution improves the viewing QoE from 0.454 to 0.438, based on slight improvements of the maximum quality level and its associated delay.

5.4.5 *Validation and Comparison*

This section aims to validate our results by comparing them to some of the corresponding works from Section 3.3.
It complements the Section 4.6 with regard to the transmission approach and the resulting viewing QoE. We showed in that section how Raffelsberger and Hellwagner [81] can achieve playout delay and quality that are similar to our result. However, our results reflect a better viewing QoE because of the maximum delivered quality.

In this chapter, we succeeded to achieve a better maximum quality level at a lower delay, leading to a still better viewing QoE. Similarly, the same comparison applies for the work of Wu and Ma [84], where our better analyzed viewing QoE is shifted forward in this chapter. Furthermore, the adaptive solution in this chapter introduces improvements of the maximum quality level and its associated delay, which helps to overcome the advantage of Chen et al. [82, 83] with regard to the viewing QoE's p_2 result point.

Besides, addressing the comparison with regard to adaptive decisions in OppNets, a different analysis must be drawn. Schierl et al. [74] adapt the channel coding replication factor by studying the quality of the connection. Whereas in Mobile Ad-hoc Networks (MANETs) it is very easy to determine the quality and stability of a connection, this is possible in OppNets only using estimations. Tournoux et al. [98] tried to perform such an estimation by analyzing the number of contacts of the encountered nodes to assign them a "node degree" value. Hence, they could build an off-line hash-map to optimize the redundancy factor of the SnW routing protocol based on given node degree and target delay. Nevertheless, using the channel quality estimation and the node degree metric for the adaptive size tuning has two main drawbacks: 1) The adaptive measurements can take place only when you are in contact with node, to which the data must be sent. Therefore, a long part of the contact duration is spent on calculating and preparing the chunk to be sent. Whereas our adaptive solution can make use of the time between contacts to update history and prepare the measurements correspondingly. 2) The off-line hash-map is based on a metric (i.e. the number of contacts) that is not directly relevant to estimate the optimal chunk size. Hence, the problem of delivery failure due to big a chunk size will arise again.

The PRoPHET routing protocol [33] also functions adaptively through its estimation function. However, PRoPHET was experimentally tested in Chapter 4, and it reported bad results in Figures 4.14 and 4.16.

Lastly, Hu and Hsieh [100] and Wang et al. [32] developed algorithms to determine whether to send data to another node upon contact or not. However, using their techniques for the adaptive size tuning will run into the above mentioned drawbacks of using the technique of Tournoux et al. [98].

5.5 SUMMARY

This chapter proposed advancements and refinements of the video dissemination scheme in OppNet. The transmission units are tuned to form chunks encapsulating the small NALUs. Those chunks vary in size according to adaptive node-dependent measurements that respect the changing network conditions of OppNets. The theoretical propositions are promoted through experimental simulations. Constantly to the

former results in Chapter 4, the 100% ratio of playout availability is kept, a slight improvement for the delivery delay is achieved, and the average received quality level is improved. Consequently, these results led to an improved value of the viewing QoE.

* * *

Chapter 6

THE INTEGRATION OF BALCON: A BACKWARD LOSS CONCEALMENT MECHANISM

6.1 PROBLEM STATEMENT

It is seen from Chapters 4 and 5 how the dissemination scheme implemented two related techniques for delivery enhancement at the sender's side. However, this did not suffice to completely eliminate cases with limited delivery. Failures in delivering certain layers are detected to be caused by a partial loss of only a small fraction of the composing NAL Units (NALUs). Such small data gaps of partial loss can usually be filled using loss- or error-concealment techniques. However, under the lack of a loss concealment technique and because of the continuity of the video medium, a Scalable Video Coding (SVC) layer is successfully delivered only if all its composing NALUs are delivered.

Therefore, this chapter addresses the third research problem (from Chapter 1), which aims to induct a concealment mechanism at the destination node. This will mitigate the impact of partial losses and enable the receiver to react to data loss.

As given in Chapter 3, most of the interest in loss- or error- concealment in OppNets is concerned with "forward" techniques, where the source node has to spend more processing power in order to calculate correction codes. These codes are attached and transmitted as a part of the original data to help the receiver node in recovering lost chunks. Some discussed drawbacks of these techniques are: the extra data to be computed and transmitted, the prolonged delivery delay and the need to modify the encoding process to integrate their implementations. Furthermore, some correction codes are sensitive to the amount and pattern of the occurred losses. That is, if an essential code part is missing, the recovery process is suspended with the receiver node sitting on his hands.

Hence, this chapter studies how to push the destination node at the back-end into action by reacting to missing data in a "backward" solution. First, the receiver asks

for the missing data from other nodes in the network in the form of request-response. Second, since the transmission is concerned with video content, video Frame Loss error-Concealment (FLC) techniques can also be exploited at the receiver side. Consequently, we propose to combine the two techniques in a composite *Backward* loss concealment solution, which can react to a limited amount of loss.

6.2 OBJECTIVES AND CONTRIBUTIONS

This section highlights the objectives and contributions of this chapter. First, it overviews the proposed loss concealment mechanism, and how the composing techniques are integrated within the frame of one concealment mechanism. Then, it discusses the criteria for initiating the concealment process.

6.2.1 *Backward Loss Concealment*

In this section, a "backward" solution is proposed, which is based on involving the receiver node in the dissemination process by reacting to missing data parts[1]. The solution is shortly named Backward Loss Concealment (BALCON), which compositely combines two concealment techniques:

1. Network demands: the receiver node asks for the missing parts from other nodes in the network in the form of request-response.
2. Video-based: because the transmission is concerned with video content, video FLC techniques can also be exploited at the receiver side.

6.2.1.1 *Network Demands*

As introduced, the first part of the composite concealment mechanism is to initiate demands by the receiver in the network, in search for the missing data parts from other nodes. Unlike Mobile Ad-hoc Network (MANET)'s Automatic Retransmission Request (ARQ), demands are not generated as acknowledgments (automatically and for every message), and their destination is not limited to the original source.
So, from now on we will refer to the receiver node in this context as the "demander" who initiates demands. Those demands can target then through multihop paths one or more nodes, whose ability to serve a demand by having a copy of the sought data part is probable. We refer to those potentially serving nodes as the "potentials". Hence, if any potential can serve the demand, it will reply and address the sought data part back to the demander.
The ratio of concealed loss by the received data parts in response to the network

1 We use "data part" as a general term that can have difference references (e.g. NALU, chunk, layer, .. etc.) depending on the context, where the concealment mechanism is applied.

demands represents the capability of the network-demands concealment technique. This capability depends on many parameters that are related to the network configuration, like the routing protocol, the mobility and the density.
However, more detail on how the demands are initiated, and how the responses are routed back, are given in Section 6.3.

6.2.1.2 *Video Frame Loss Error Concealment*

Video content-based concealment techniques represent another possibility at the receiver side, and hence can complement the network demands in the composite mechanism. As given in Section 3.4, the SVC reference software implementation (Joint Scalable Video Model (JSVM) [2]) adopts four basic concealment techniques: Frame copy, Temporal direct, BL-skip and Reconstruction BL-upsampling. All the four mentioned techniques are mainly designed to conceal errors in the Base Layer (BL) as well as the temporal and spatial Enhancement Layers (ELs). Their applicability for the Signal-to-Noise Ratio (SNR) quality ELs has to go through a more complex process, which is a different research topic [123]. Therefore, this is kept out of the concern of this work.
The four FLC techniques are already implemented and tested in the literature [106, 107]. Hence, their concealment capability can be easily evaluated by measuring the associated video quality drop (in relevance to the quality[2] of the original video) when they are applied on on different loss ratios. Then, fixing a certain value, below which the video quality is not acceptable, will refer to the concealable amount of loss by the corresponding video FLC technique.

6.2.2 *Launching Criteria of The Concealment Process*

We propose to calculate the launching moment or criteria of the whole concealment process by setting a threshold on either time or delivery ratio.

6.2.2.1 *Time*

In this approach, BALCON starts after a time T_{start}, that is measured from the starting point of the dissemination. The concealment process is launched then regardless of the delivery and loss ratios. This criteria is easy to implement because it does not involve any computation and the value of T_{start} can be directly set either arbitrarily or based on a certain delay value. For the latter case, when the delivery delay exceeds a predefined threshold, the concealment mechanism is launched.
In both cases, the ignorance of the present amount of loss makes the concealment mechanism prone to one of two deficiencies: either to start too early, then the con-

2 Quality is measured by the means of Peak Signal-to-Noise Ratio (PSNR) metric.

cealment can not help to complete the delivery. Or to start too late, then an avoidable delivery delay will take place.

6.2.2.2 *Delivery*

In the second variation, BALCON starts when the delivery ratio exceeds a fixed value, regardless of when that would happen. The computation of the threshold value has to be done precisely, with respect to the loss and the concealment capabilities of the used techniques, so that:

$$\underbrace{data_{total} - data_{delivered}}_{\text{loss}} \leqslant concealment_{demands} + concealment_{FLC}$$

This is to be interpreted as: the total concealment capability must be no less than the present amount of loss, which ensures a feasible application of BALCON.
Deriving the launching threshold can be done then by knowing the capabilities of the concealment techniques, and waiting for the parameter ($data_{delivered}$) to increase until it satisfies the given relation. This is explained in more detail later in Listing 6.1.

6.3 NETWORK DEMANDS: REQUESTS & RESPONSES

This section addresses in more details the network demands technique, through its two phases: the initiation of demands, and serving them back.

6.3.1 *Requests: Initiating Demands*

The set of target potentials can include the original source node only (Unicast), further nodes that are known to the demander (Multicast), or any encountered node in the network (Broadcast). Thus, the distribution is subject to a trade-off between the probability of serving the demand, on the first hand, and the privacy[3] and resources consumption on the other hand.
The probability that any single potential can serve the demand is named "servability," which is (for any potential rather than the source) tightly dependent on the originally used routing protocol and the resulting distribution of data among the mobile nodes. For instance, if a message is distributed using the Spray-and-Wait (SnW) routing protocol with M copies, in a network having n nodes, the servability of any potential is limited to:

$$servability \leqslant \frac{M}{n} \tag{6.1}$$

3 For example, who is allowed to know which data parts are missing? However, this is a different research topic, and it is kept out of the scope of this work.

The upper bound applies after a time T, when the M copies are equally distributed to M nodes, leaving no node with more than one copy of the message. At any time $t < T$, only $m < M$ nodes are able to serve.
All of above constitutes the case when only one potential is targeted. Then the servability when targeting k potentials is:

$$\begin{aligned} servability_{(k)} &= P(\text{at least one potential can serve}) \\ &= 1 - P(\text{none of the } k \text{ potential nodes can serve}) \\ &= 1 - P(node_1 \text{ can't } \& \ node_2 \text{ can't } \& \ .. \ \& \ node_k \text{ can't}) \end{aligned}$$

The probability that a potential can not serve the demand is given by the ratio of the number of nodes that do not have a copy of the message, to the total number of nodes. Hence, because the events of that the k potentials can not serve the demand are not mutually exclusive, then:

$$servability_{(k)} = 1 - (\underbrace{\frac{n-m}{n} \times \frac{n-m-1}{n-1} \times ... \times \frac{n-m-(k-1)}{n-(k-1)}}_{k \text{ times}})$$

$$\Rightarrow servability_{(k)} = 1 - (\frac{\sigma(n-m,k)}{\sigma(n,k)}) \tag{6.2}$$

Where $P(..)$ refers to probability, and $\sigma(..)$ refers to permutation. Obviously, when targeting more potentials (greater k), the servability would be higher, i.e. the probability that the demand will be served by any potential, regardless of its identity.

6.3.2 *Responses: Serving Demands*

When a potential receives a demand of a specific data part, it looks that part up in its buffer. If found, the demanded part is routed back to the demander. For this, the potential can use either the originally used routing protocol, or configure a new protocol.
In Chapters 4 and 5, the SnW routing protocol was used with different redundancy factors for the different data parts depending on their importance degree. In this case, if a potential receives a demand of an existing part, the demand can act then as a trigger to increase the redundancy factor of that part, which will allow to initiate more copies and help to serve the demand back. For example, the source initiates a message with c_i copies. This message reaches later a node X with a left forward-allowance (redundancy factor) of only one copy. This copy must reside at X until it can be directly transmitted to its destination. Now, if X receives a demand for this message, X will reset its redundancy factor from 1 to c_i again to increase its delivery

probability. Otherwise, a newly configured routing protocol can be independent of the first one.

6.3.2.1 *The Applicability of Smart Routing*

In less dynamic peer-to-peer networks, the path of the demand message along the way from the demander to the potential, might represent a useful information for routing back response messages. These latter can, for example, try to go through the same or a very similar path to the former one. In Opportunistic Networks (OppNets), on the other hand, this is hardly applicable because of the high dynamicity of the network. For example, assume a node A delivers a message to node B, and they both continue their movements. Thus, a disconnection is highly probable, so that B will no longer be able to send a response back through A.
To investigate this assumption practically, we performed a "ping-pong" message exchange experiment in a simulated OppNet environment between 100 different and random node pairs[4]. So that a random node x_i sends a very small message (1 byte) to a random node y_i using the epidemic routing protocol. When delivered, y_i returns the same message back to x_i on any path using the same routing protocol. To study the correlation between the path of any of the messages and their responses, we counted the common nodes between the path of a message and the path of its response. Among over 500 exchanged messages, we got the following results:

- Number of hops $\in [1, 7]$, $Avg \approx 3$
- $\sim$ 49.5% of paths had 0 common node
- $\sim$ 50% of paths had 1 common node
- Only $\sim$ 0.5% of paths had 2 common nodes

Consequently, it is concluded to let response messages make no prior-knowledge assumptions for the routing protocol, because targeting any of the nodes on the demand path as a next relay, would unpredictably delay the delivery if that node is far or left the network.

6.4 THE FUNCTIONALITY OF BALCON

Our proposal is to combine the two previously described techniques in a composite solution (i.e. BALCON) that can work better at the receiver side. As seen in Chapter 5, a partial loss of a small amount of the NALUs dramatically affects the total delivery. Therefore, the goal of BALCON is to conceal the loss of missing NALUs.

4 Other setup parameters are the same as used for the experiments in Chapter 4 and 5: SPMBM mobility, Bluetooth transmission interface, a big enough buffer and a long enough TTL.

For a specific SVC Operating Point (OP), BALCON applies both network demands and FLC techniques in an adaptive way with regard to the individual types of the missing NALUs (i.e. BL, temporal-, spatial- or SNR-enhancing unit). The designated functionalities for the different NALU types are summarized in the algorithm in Listings 6.1 and 6.2.

```
Input:
   Delivered data: delivered
   Delivery threshold: D
   Amount of loss that can be concealed using FLC: C_FL

% Check the delivery ratio with regard to the defined threshold
while (delivered < D) {
   wait();
}
% When the delivery exceeds the defined threshold, BALCON starts

% Determine the missing NALUs and apply the network-demads technique
missingNALUs = recognizeMissingNALUs(delivered);
delivered += networkDemands(missingNALUs, C_FL);

% Determine the still missing NALUs and apply the FLC technique
missingNALUs = recognizeMissingNALUs(delivered);
for (x in missingNALUs) {
   % The functionality varies depending on the NALU  type
   if (x.type == BL || x.type == EL-Temporal) {
      delivered += FLC(delivered, missingNALUs, "Frame-copy");
   }

   else if (x.type == EL-Spatial) {
      if (BL.exists) {
         delivered += FLC(delivered, missingNALUs, "Frame-copy");
      }
      else {
         delivered += FLC(delivered, missingNALUs, "BL-skip");
      }
   }

   else if (x.type == EL-SNR) {
      % No FLC is applied
   }
}
```

Listing 6.1: BALCON Functionality - 1

Because FLC is based on extrapolating missing NALUs, it has more impact on the quality. Hence, it is chosen to always start with the network demands technique, and then apply FLC on the remaining loss fraction. Furthermore, the launching criteria is set for BALCON to "delivery," because the whole mechanism is based on analyzing the

```
function recognizeMissingNALUs(delivered) {
   allNALUs = delivered.metadata.listOfNALUs;
   missingNALUs = [];
   for (x in allNALUs){
      if (delivered.hasNALU(x) == false){
         missingNALUs.push(x);
      }
   }
   return missingNALUs;
}

function networkDemands(missingNALUs, C_FL) {
   servedDamends = [];
   % Send demands
   for (x in missingNALUs) {
      broadcastDemand( x );
   }
   % Wait for responses
   while (missingNALUs.size > C_FL) {
      if (new NALU, x, is received) {
         servedDemands.push(x);
         missingNALUs.delete(x);
      }
      % else wait
   }
   % The function returns only when the amount of the remaining loss can be
       concealed using FLC
   return servedDemands;
}
```

Listing 6.2: BALCON Functionality - 2: Utility Functions

capabilities of both concealment techniques. The algorithm waits for the loss ratio to drop under a specific threshold (or the delivery ration to go over a specific threshold), so that it can be concealed then. Determining this threshold is tightly dependent on the mentioned capabilities. It starts by studying the capability of the FLC technique by measuring the video quality drop associated with different loss ratios, as presented in Section 6.2.1. Next, the capability of the network demands approach is measured as a percentage of the demands that can be successfully served. Since the value of the capability must be known by the algorithm, there are two possibilities to measure it:

- Statically: by performing off-line experiments to estimate the capability value of the network demands technique, then fix this value statically for further experiments.
- Dynamically: by making BALCON self-adaptive, so that to take the network conditions into consideration and calculate the capability at run-time.

To concentrate on the concealment aspect of the algorithm, we limit the experiments in this chapter to the statical approach. However, the self-adaptive BALCON mechanism is kept as a future work.
Lastly, unserved demands represent the remaining loss fraction, which must be equal to the capability of FLC.
Consequently, by merging the two approaches together, a threshold for the whole BALCON solution can be defined, at which the concealment process will start. Mathematically formulated, BALCON should satisfy the following relation:

$$l - (C_{ND} \times l) = C_{FL} \tag{6.3}$$

where

l : total amount of loss
C_{ND} : ratio of servable demands, $\in [0, 1[$
C_{FL} : concealable loss using FLC

Rearranging the relation, it can be written as:

$$l = \frac{C_{FL}}{1 - C_{ND}} \tag{6.4}$$

Where l in this last relation represents the loss threshold, with which the concealment process may start.
Lastly, this is repeated for each OP separately, because the OPs (and their constituting layers) posses different scalability characteristics and importance degrees, which results in varying the threshold values.

6.5 EXPERIMENTS

This section presents and analyzes the results of executed experiments. Again for the continuity of work, the experimental setup is remained untouched from Section 4.3. Moreover, as in Chapter 5, the tested video sequence is limited to V_2: *Highway*, the routing protocol to *binary SnW*, and the mobility model to *Shortest Path Map-Based Movement (SPMBM)*.

6.5.1 *Capability Estimation of The Network Demands*

The first set of experiments is meant to approximate the capability of the network demands technique in general, out of the context of video data.
Messages that are similar in size to NALUs are transmitted in the simulated network

toward the receiver node, which will turn into a demander after an arbitrarily fixed time T_{start} regardless of the delivery. Responses are analyzed in order to statistically set the parameter C_{ND} in Equation 6.4. Epidemic and SnW (with different numbers of copies) routing protocols were tested to route responses back. SnW copies are identical to the number of copies that were already used to transmit the SVC OPs in previous chapters, namely: 16, 8, 4 and 2 copies. Besides, demands have two options with regard to the target group: one or multiple potential nodes. Results for each case are described separately. Measurements include the ratio of returned responses to the demander, and their delays.

6.5.1.1 *One Potential Node*

To limit the resources consumption and implement an easier selection of target potentials, the demander sends the demands exclusively to the source node, which for any successfully received demand has a servability of 100%. Results for epidemic and the different SnW variations are shown in Figure 6.1.

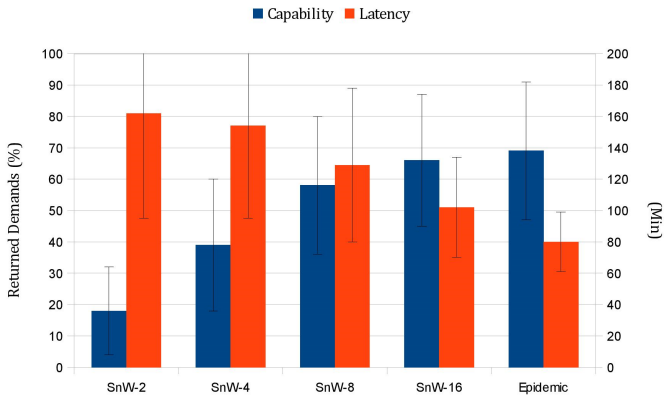


Figure 6.1: Network Demands: One Potential

Using SnW with 2 copies, only 18% of the demanded messages could have been successfully received back, with an average delay of about 160 min. Results become better while using more copies, until reaching 66% of successful reception and $\approx$ 100 min delay using SnW with 16 copies. Lastly, using the uncontrolled flooding, Epidemic reports 69% of successful reception and 80 min delay.

6.5.1.2 *Multiple Potential Nodes*

With less strict constraints, demands can target a larger set of potentials. Due to the characteristics of OppNets, the demander does not know in advance the member nodes of the network. Therefore, to expand the set of potentials, demands will be broadcasted in the network. Results for epidemic and the different SnW variations are shown in Figure 6.2.
Results in Figure 6.2 are obviously better than those in Figure 6.1 in terms of both

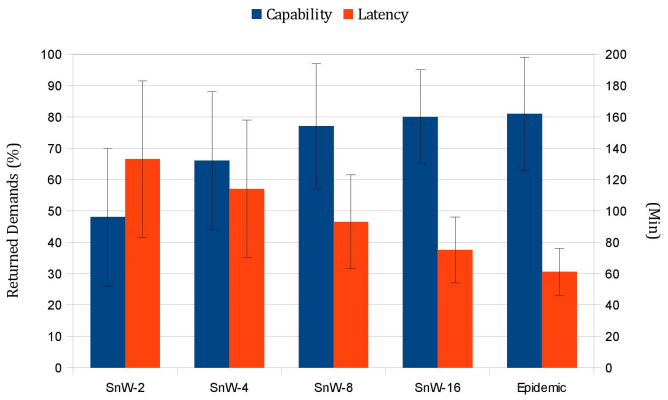


Figure 6.2: Network Demands: Multiple Potentials

successful reception of demands and delay. Using SnW with 16 copies reports a successful reception of 80% and around 70 min delay. This means that with targeting multiple potentials using the SnW-16 routing protocol, the network-demands technique is capable to conceal 80% of existing loss.
These results will be used next to derive the capability and impact of applying BALCON.

6.5.2 *Concealment Capability of BALCON*

The proposals detailed in Section 6.4 are now put into practice, relying on already existing evaluations in the literature to measure the FLC capability. The concealment capability for each NALU type is be measured, and then all concealment techniques are combined together to analyze the impact of BALCON.

6.5.2.1 *BL & Temporal Enhancing*

BASE LAYER Keränen et al. [107] evaluated many video FLC techniques on different loss patterns. For example, when using frame-copy for the concealment on 10% loss ratio, the PSNR value drops to about 50% of its error-free original value. Now to determine the concealable amount of loss by the frame-copy technique, the minimum acceptable video quality drop must be set. For instance, if 75% of the PSNR value is taken as an acceptable one, it will hold $C_{FL} = 5\%$.
Furthermore, for the BL, the SnW routing protocol is associated with 16 copies as presented before. Hence, from Figure 6.2 we can extract that $C_{ND} = 0.8$.
By applying these values into Equation 6.4, the threshold loss value will be then:

$$l_{threshold} = \frac{0.05}{1 - 0.8} = 0.25 = 25\%$$

This is interpreted as: when the loss ratio is no more than 25%, the concealment process for the BL NALUs starts, with the network demands first and FLC then.

TEMPORAL ENHANCING The BL and temporal enhancing units are subject to the same video FLC evaluation: $C_{FL} = 5\%$. Besides, for the temporal EL, the SnW routing protocol is associated with 8 copies as presented before. Hence, from Figure 6.2 we can extract that $C_{ND} = 0.76$. By applying these values into Equation 6.4, the threshold loss value will be then:

$$l_{threshold} = \frac{0.05}{1 - 0.76} = 0.208 = 20.8\%$$

This holds that when the loss ratio is no more than 20.8%, the concealment process for the temporal enhancing NALUs may start, with the network demands first and FLC then.

6.5.2.2 *Spatial Enhancing*

Chen et al. [106] evaluated the FLC technique on wider loss patterns including more NALUs types. Loss in spatial-enhancing NALUs is less sensitive than in the BL, however the concealment is affected by the loss on both levels. Using an inter-layer technique (BLSkip) on a loss of 10% in the spatial EL drops the PSNR value of the concealed video to about 85%. However, this technique assumes that the corresponding BL frame is correctly present. The same loss causes a PSNR drop equal to 75% of its original value when using an intra-layer technique. By accepting these values as the minimum acceptable video quality drop, we can set $C_{FL} = 10\%$.
For the spatial ELs, SnW uses 4 copies. So from Figure 6.2 we find $C_{ND} = 0.66$, which is to be applied into Equation 6.4:

$$l_{threshold} = \frac{0.1}{1 - 0.66} = 0.294 = 29.4\%$$

As before, when the delivery allows to get $< 29.4\%$ loss, the concealment process for the spatial enhancing NALUs may start.

6.5.3 *The Complete Functionality of BALCON*

6.5.3.1 *Impact on SVC*

To clearly show the impact of BALCON, it is more efficient to build on the experiment from Section 5.4.1, where NALUs are transmitted in an OppNet scenario. We recall results from Figure 5.3 again in Figure 6.3.
The shown high delivery failure is mainly because of the partial loss in the big set of composing units, causing whole layers to be useless. Only a playout availability of 58% could have been concluded, with a dominant limited BL quality. Whereas 42% of the simulation rounds had no video playout at all.
By analyzing the cases of an undelivered BL (no playout) in that experiment, it is found that failures were caused by 10-20% partial loss. That is, for a given example

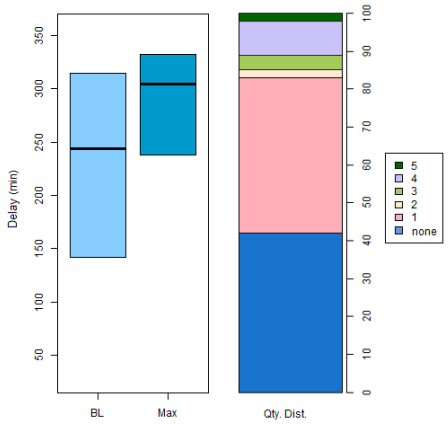


Figure 6.3: Transmission of NAL-Units

undelivered BL, 80-90% of its composing NALUs are reported to be successfully delivered. But without a concealment mechanism, this small fraction of partial loss leads the delivery of the corresponding BL to fail. Similarly, limited higher quality is caused by about 30% partial loss in higher OPs. An example of an undelivered BL and an undelivered EL is depicted in Figure 6.4 as a progress over time.

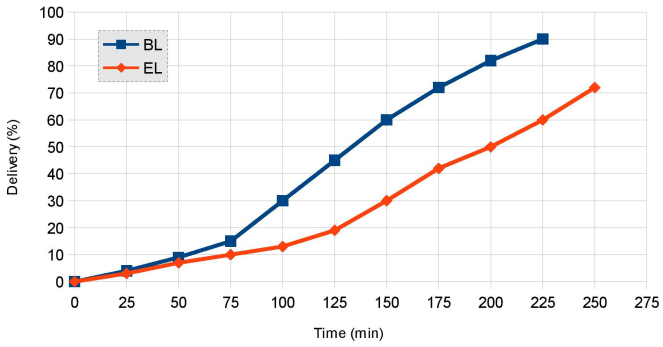


Figure 6.4: Delivery Progression of an Undelivered BL and an Undelivered EL

Time instances when the concealment process starts, can also be derived from Figure 6.4. As calculated in Section 6.5.2, $l_{threshold} = 25\%$ for the BL. This is equivalent to a delivery of 75%, which is reached around the 180^{th} minute. For the spatial EL, $l_{threshold} \approx 30\% \Rightarrow delivered \approx 70\%$, which corresponds to the 240^{th} minute.
Consequently, after experimentally applying BALCON, new results are reported in Figure 6.5.
Compared to Figure 6.3, the quality distribution is enhanced to reflect a 100% playout availability and an upgraded quality level. But on the other hand, results regarding the delay show no refinement, because of the extra delay generated by the network demands approach.
Nevertheless, since we take the statical approach to calculate the threshold values in the experiments, these values can not be generalized. The capability thresholds dif-

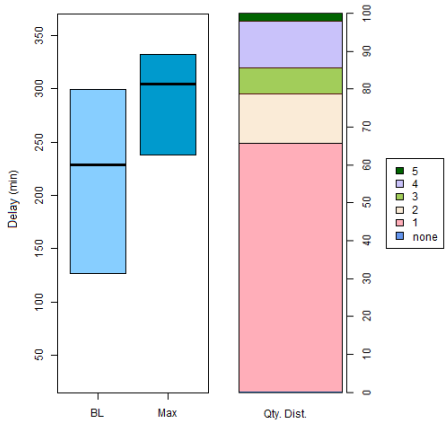


Figure 6.5: Transmission of NAL-Units + Applying BALCON

fer from one application to another, because they depend on many parameters like: video concealment technique, video content, desired PSNR values and network settings. However, the statical measurement serve in the presented experiments to show the applicability and functionality of the BALCON mechanism in our video dissemination scheme in OppNets.

6.5.3.2 *Varying The Starting Point*

This section will study the influence of starting the concealment process earlier or later than the computed threshold.
For instance, for the BL case in the last experiments, the computed loss threshold is: $l_{threshold} = 25\%$. By letting BALCON start at this threshold value, we argue that it will conceal all of the loss and reach a full delivery ratio, because the amount of loss corresponds to the concealment capability of BALCON (i.e. 25% as calculated in Section 6.5.2.1). Now, if BALCON starts earlier, for example when $delivery = 70\% \Rightarrow loss = 30\%$: by adding the BALCON capability, the delivery ratio may reach 70+25=95%. Hence, the full delivery ratio could not have been reached. Moreover, as Equation 6.2 suggests, when starting earlier the distribution of the m messages by the SnW routing protocol will not reach the upper limit M. Therefore, only $m < M$ nodes will have copies then, which holds a lower servability value and hence also a lower C_{ND} capability.
On the other hand, starting later at a delivery level higher than the computed threshold is not feasible, because either the higher level will not be reached, or if reached there will be no benefit but only a higher delay. For example, if BALCON waits until $delivery = 80\% \Rightarrow loss = 20\%$, then (if this level is reached) BALCON is already more capable than the remaining amount of loss.
Consequently, starting the concealment process at arbitrary points is not recommended, because either a full delivery ratio will not be reached, or an extra delay will be wastefully added.

6.5.3.3 *Validation and Comparison*

To the best of our knowledge, loss recovery in OppNets is mainly addressed by applying network and channel coding techniques, as already given in Section 3.4.
Nguyen et al. [110] use Reed-Solomon Forward Error Correction (FEC) on a loss ratio of 20%. The recovered videos report PSNR drops of 77-79% of their original values. Whereas BALCON reaches better quality levels at higher loss ratios, i.e. 75% PSNR quality at 25% loss for the BL, and 75-85% PSNR quality at 20.8-29.4% loss for higher enhancing levels. Besides, with regard to the delay results, no direct numerical comparisons can be drawn because of the many different setting parameters. Instead, we try to estimate the results as in Section 4.6. To reach a playout availability ratio of 100%, the authors technique takes longer because of the prolonged distribution of packets by the source. However, BALCON also involves extra delay through the network-demands technique, since it waits for the demands to be distributed and the responses to be received back.
A better perceived quality could have been achieved by Schierl et al. [73], and a lower delay by Sardari et al. [109], both by using multiple sources to distribute the FEC code blocks. However, this is not applicable in our case, because of the need for a coordination between the source nodes, which is not available in OppNets.

6.6 SUMMARY

This chapter is intended to implement and study the feasibility of a backward loss concealment mechanism (shortly named as BALCON) for scalable videos in OppNets. We proposed a solution at the receiver side, that is able to conceal loss below a specific threshold. BALCON applies a composite concealment solution, that is based on video Frame Loss error-Concealment (FLC) techniques as well as network demands. Experiments of an example scenario showed the applicability and feasibility of our proposal. In the experimented example, BALCON was capable of concealing a loss amount of 25% in the BL and 20.8%-29.4% in higher levels. However, these derived threshold values in the experiments can not be generalized, since they may differ from one application to another. Threshold values depend on many parameters like: video concealment technique, video content, desired PSNR values and network settings, under which there exist many other parameters. Nevertheless, in sparse environments and applications that accept some quality drop, the BALCON mechanism can successfully be applied to conceal an amount of loss similar to the results of our example experiment.

* * *

Part III

CONCLUSION

Chapter 7: Conclusion and Future Work.

Now this is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning.

— Winston Churchill

Chapter 7

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis by first overviewing the defined context of work and the addressed issues. Then, it highlights the made contributions. And lastly, it suggests possible directions of further improvements and future research.

7.1 OVERVIEW

Through the massive growth of handheld devices both quantitatively and qualitatively, a new ad-hoc networking pattern has been enabled, by which different devices may communicate, interact and share content. The general idea behind ad-hoc networking is to offer a decentralized connection, which decouples the communicating nodes from access points and base stations. When two nodes are physically close enough to be in the communication range of each other, they can substitute the data exchange through a centralized infrastructural entity with an ad-hoc short-range alternative. A Mobile Ad-hoc Network (MANET) communication might have a longer path than a direct source-destination data delivery. Data may hop through multiple intermediary nodes to reach the destination. Furthermore, Opportunistic Networks (OppNets) induct more tolerance into the network functionality with regard to nodes mobility and resulting disconnection. Using the Store-Carry-Forward (SCF) paradigm, intermediary nodes are allowed to store received messages and carry them until having the opportunity to forward them again.
Consequently, OppNets support indirect end-to-end data delivery and possess a dynamic network topology. Nevertheless, because forwarding opportunities depend on nodes' mobility, communication disruptions and uncertain data delivery are dominant. These characteristics made the usual use case scenarios of OppNets limited to the exchange of small messages, in relation for example to traffic notifications or disaster alarms. However, the available capabilities of the handheld devices created possible scenarios to share User-Generated Content (UGC) of multimedia, where users need to lay infrastructural means aside and switch to a multihop ad-hoc alternative. Examples are the cases of absence, high costs or undesirable censorship of the infras-

tructural networks.
The challenge of a video dissemination application is that OppNets would expose large-sized video data to partial loss, which is assumed to make any received parts useless because of the continuity of the medium. Despite the impossibility of realizing a streaming application with hard time constraints in this context, we propose a dissemination scheme that supports the *"download-and-play"* use-case. This scheme is made up of two solution folds:

- Delivery enhancement at the sender's side, depending on data partitioning and adaptive data volumes tunning.
- Loss concealment at the receiver's side to overcome small fractions of partial losses.

The objective of the scheme is an enhanced video viewing experience, by the means of playout delay, delivery ratio that determines the maximum achievable quality, and the overhead that is applied on the network.
The Opportunistic Network Environment (ONE) simulator is configured to experiment the proposed solutions in a simulated environment. A series of experiments are preformed, concluding through the gathered results the applicability of the proposed video dissemination scheme within the frame of OppNets.

7.2 CONTRIBUTIONS

The thesis presented three contributions in Chapters 4, 5 and 6 for the proposed video dissemination scheme.

FIRST, the scheme applies the Scalable Video Coding (SVC) technique, which divides videos into scalable layers of smaller volumes, holding different levels of quality. The dependency among layers marks them with multiple unequal importance degrees. This property is exploited to prioritize the video layers and route them with unequal degrees of redundancy proportionally to their importance. Consequently, playout delay is reduced by stressing on faster and more reliable delivery of layers with lower quality (but higher importance).
Furthermore, an evaluation metric for the viewing experience is defined, involving the delivery delay, video quality and network overhead.

SECOND, the scheme goes deeper in the direction of data partitioning, by not sticking to the scalable layers with predetermined sizes. Instead, data volumes are tuned adaptively with regard to the continuously changing network conditions of OppNet. Thus, the scheme enables each node to record a history of environmental information, and use this history then to predict future forwarding opportunities, and optimize the data volumes accordingly.

THIRD, the scheme involves the destination node by implementing a feasible concealment mechanism at its side, with the capability to react to an amount of loss. The implemented concealment mechanism functions in a composite manner with a twofold profit: a content-based video frame loss concealment, and a network-based loss concealment through demands, that are initiated and cast in search for the missing units from other nodes in the network.

* * *

7.3 FUTURE WORK

Due to many constraints, for instance in time and deployment, it is hard to meet perfection in research. Consequently, many aspects in the research work remain untouched or imperfect. Therefore, this section presents future visions of the thesis, which can promise further improvements or open the door for new domains of research.

7.3.1 *Further Improvements*

Future prospects to further improve this work can be observed from the perspective of the identified contributions in Chapter 1.

C_1 & C_2 The first two contributions focused on delivery enhancement at the sender's side, depending on data partitioning and adaptive data volumes tunning. Aspects that could have been addressed in a better or more advanced way include:

- Real testbeds or real mobility traces: Despite their high costs, the real testbeds are the most reliable approach to perform experiments and collect real-world-driven statistics. However, under the difficult applicability of real testbeds, using real mobility traces helps to give an insight about how the proposed solution would look like when actually applied.
- Dynamic selection of Operating Points (OPs): The selection of OPs to overlay the SVC layers (as given in Section 4.3.1.4) can be done in a more advanced and dynamic way with regard to the resulting viewing experience, based on the size and the video content.

C_3 The last contribution tackled a loss concealment mechanism at the receiver's side to overcome small fractions of partial losses. The functionality of the mechanism is based on static calculations to measure the concealment capabilities of the used techniques, as given in Section 6.4. Thus, a self-adaptive loss concealment mecha-

nism is very encouraged to be addressed. That is, the mechanism can estimate those capabilities at run-time, based on the video content and the network conditions.

7.3.2 *New Domains of Research*

Beside further improvements of already existing topics, complete new domains of research can also be initialized as extensions to this work. Examples basically include:

- Security: In OppNets, security is a very important and wide topic, and can be addressed from different aspects, as for instance:
  - Privacy: The identity of the communicating nodes must be kept hidden at the communication interface, as well as within the exchanged content.
  - Forgery: Prevent unwanted or irrelevant content from flowing in the network.
- Hybrid (mesh) networks: It is very interesting to study the impact of the proposed solutions when applied in a network that can switch between the OppNet and the infrastructural communication domains. The advantages of both types of networks can then combined, and new challenges are introduced, like:
  - Data distribution and consumption control: Decide for which network to use when sending specific data.
  - Further security challenges, when using an unreliable infrastructural network.

* * *

Part IV

APPENDIX

Appendix A: Technical Details

The most important property of a program is,
whether it accomplishes the intention of its user.

— C.A.R. Hoare

Appendix A

TECHNICAL DETAILS

This appendix overviews the essential technical material that enables delivering the scientific contribution of this thesis. The technical material includes instructions, implementations and techniques from the following addressed aspects:

- Scalable Video Coding (SVC) encoding and decoding
- Computations with regard to video quality
- Functionality adaptation of the open-source simulator

A.1 SVC ENCODER/DECODER

This section highlights how to configure and SVC-encode a raw video using the Joint Scalable Video Model (JSVM) tools. As an input for the examples below, we consider a raw video file "`bus_352x288_30.yuv`", which has a spatial resolution of 352×288 and a frame rate of 30 frame per second (fps).

A.1.1 *Encoder*

Using the JSVM library, SVC encoding is simply performed from the command line using the following instruction:

```
> H264AVCEncoderLibTestStatic.exe -pf main.cfg
```

Listing A.1: SVC Encoding Instruction

Where the whole configuration has to be in the text file "`main.cfg`". An overview of the used main configuration file is given in Listing A.2.
It is noticed that the main configuration file is fed with other "`.cfg`" files, which determine the spatial and Signal-to-Noise Ratio (SNR) scalability layers in the the output video file "`coded.264`", whereas the number of temporal layers is dependent

```
OutputFile              ..\vid\bus\coded.264   # Bitstream file
FrameRate               30.0       # Maximum frame rate [Hz]
FramesToBeEncoded       150        # Number of frames (at input frame rate)
GOPSize                 4          # GOP Size (at maximum frame rate)
CgsSnrRefinement        1          # SNR refinement as 1: MGS; 0: CGS
EncodeKeyPictures       1          # Key pics at T=0 (0:none, 1:MGS, 2:all)
MGSControl              1          # ME/MC for non-key pictures in MGS layers
                                   # (0:std, 1:ME with EL, 2:ME+MC with EL)
BaseLayerMode           0          # Base layer mode (0,1: AVC compatible,
                                   #                  2: AVC w subseq SEI)
SearchMode              4          # Search mode (0:BlockSearch, 4:FastSearch)
SearchRange             32         # Search range (Full Pel)
NumLayers               4          # Number of layers
LayerCfg                ..\vid\bus\layer0.cfg # Layer configuration file
LayerCfg                ..\vid\bus\layer1.cfg # Layer configuration file
LayerCfg                ..\vid\bus\layer2.cfg # Layer configuration file
LayerCfg                ..\vid\bus\layer3.cfg # Layer configuration file
```

Listing A.2: SVC Encoding: JSVM Main Configuration File

on the set "`GOPSize`" parameter[1]. Namely, there will be $\log(\text{GOPSize}) + 1 = \log(4) + 1 = 3$ temporal layers for the given configuration file.

A.1.1.1 *Spatial Scalability*

To achieve spatial scalability, down-sampled versions of the original raw video must be manually generated using JSVM's "`DownConvertStatic`" tool. For instance, to down-sample the original video "`bus_352x288_30.yuv`" by a ratio of 2, the following instruction is used:

```
> DownConvertStatic.exe 352 288 bus_352x288_30.yuv 176 144 bus_176x144_30.yuv
```

Then both files "`bus_352x288_30.yuv`" and "`bus_176x144_30.yuv`" are used as input files in the complementary configuration files of the scalability layers. This is shown in Listings A.3 and A.4.

```
InputFile           ..\vid\bus_176x144_30.yuv # Input  file
SourceWidth         176            # Input  frame width
SourceHeight        144            # Input  frame height
FrameRateIn         30             # Input  frame rate [Hz]
FrameRateOut        30             # Output frame rate [Hz]
```

Listing A.3: SVC Encoding: JSVM Configuration File layer0.cfg (basic)

1 Refer to Section 2.3.2 for more details.

```
InputFile           ..\vid\bus_352x288_30.yuv  # Input  file
SourceWidth         352             # Input  frame width
SourceHeight        288             # Input  frame height
FrameRateIn         30              # Input  frame rate [Hz]
FrameRateOut        30              # Output frame rate [Hz]
```

Listing A.4: SVC Encoding: JSVM Configuration File layer2.cfg (basic)

A.1.1.2 *SNR Scalability*

The SNR scalability depends on the fixed quantization parameters in the layers configuration files. To have multiple SNR scalability layers in a given spatial resolution, the corresponding layer configuration file has to be repeated with different quantization parameters. In the last example, "`layer1.cfg`" is repeated from "`layer0.cfg`" to have a further SNR scalability dimension at the spatial resolution of 176×144. Consequently, the two complete configuration files are given in the Listings A.5 and A.6.

```
InputFile           ..\vid\bus_176x144_30.yuv # Input  file
SourceWidth         176             # Input  frame width
SourceHeight        144             # Input  frame height
FrameRateIn         30              # Input  frame rate [Hz]
FrameRateOut        30              # Output frame rate [Hz]

QP                  34              # Quantization parameters
MeQP0               32              # QP for mot. est. / mode decision (stage 0)
MeQP1               32              # QP for mot. est. / mode decision (stage 1)
MeQP2               32              # QP for mot. est. / mode decision (stage 2)
MeQP3               32              # QP for mot. est. / mode decision (stage 3)
MeQP4               32              # QP for mot. est. / mode decision (stage 4)
MeQP5               32              # QP for mot. est. / mode decision (stage 5)
```

Listing A.5: SVC Encoding: JSVM Configuration File layer0.cfg

```
InputFile           ..\vid\bus_176x144_30.yuv # Input  file
SourceWidth         176             # Input  frame width
SourceHeight        144             # Input  frame height
FrameRateIn         30              # Input  frame rate [Hz]
FrameRateOut        30              # Output frame rate [Hz]
QP                  28              # Quantization parameters
MeQP0               28              # QP for mot. est. / mode decision (stage 0)
MeQP1               28              # QP for mot. est. / mode decision (stage 1)
MeQP2               28              # QP for mot. est. / mode decision (stage 2)
MeQP3               28              # QP for mot. est. / mode decision (stage 3)
MeQP4               28              # QP for mot. est. / mode decision (stage 4)
MeQP5               28              # QP for mot. est. / mode decision (stage 5)
```

Listing A.6: SVC Encoding: JSVM Configuration File layer1.cfg

The same applies for the second spatial resolution of 352 × 288, where the configuration file "layer3.cfg" is repeated from "layer2.cfg" to have a further SNR scalability dimension at the corresponding spatial resolution. Consequently, the two complete configuration files are given in the Listings A.7 and A.8.

```
InputFile           ..\vid\bus_352x288_30.yuv  # Input  file
SourceWidth         352            # Input  frame width
SourceHeight        288            # Input  frame height
FrameRateIn         30             # Input  frame rate [Hz]
FrameRateOut        30             # Output frame rate [Hz]
QP                  38             # Quantization parameters
MeQP0               38             # QP for mot. est. / mode decision (stage 0)
MeQP1               38             # QP for mot. est. / mode decision (stage 1)
MeQP2               38             # QP for mot. est. / mode decision (stage 2)
MeQP3               38             # QP for mot. est. / mode decision (stage 3)
MeQP4               38             # QP for mot. est. / mode decision (stage 4)
MeQP5               38             # QP for mot. est. / mode decision (stage 5)
```

Listing A.7: SVC Encoding: JSVM Configuration File layer2.cfg

```
InputFile           ..\vid\bus_352x288_30.yuv  # Input  file
SourceWidth         352            # Input  frame width
SourceHeight        288            # Input  frame height
FrameRateIn         30             # Input  frame rate [Hz]
FrameRateOut        30             # Output frame rate [Hz]
QP                  34             # Quantization parameters
MeQP0               34             # QP for mot. est. / mode decision (stage 0)
MeQP1               34             # QP for mot. est. / mode decision (stage 1)
MeQP2               34             # QP for mot. est. / mode decision (stage 2)
MeQP3               34             # QP for mot. est. / mode decision (stage 3)
MeQP4               34             # QP for mot. est. / mode decision (stage 4)
MeQP5               34             # QP for mot. est. / mode decision (stage 5)
```

Listing A.8: SVC Encoding: JSVM Configuration File layer3.cfg

A.1.2 *Decoder and Player*

Although all of the experiments are performed on a simulator, and the receiving nodes receive videos only virtually, a video decoder and player is needed to test the scalability and practice the applicability of SVC. To decode and play SVC coded videos, we use the OpenSvc Decoder (OSD) [124], which is an open-source library that is built into the "mplayer" open-source video player. OSD with mplayer give the possibility at run time to change video parameters as the frame rate and resolution.
Playing a video using the mplayer can be done by executing the following instruction form the command line:

```
> mplayer coded.264 -fps 30
```

Where "`coded.264`" is the name of the encoded video file. The scalability options can be changed then during the decoding process using a set of "hot-keys" as described by Blestel and Raulet [124].

Besides, it is also possible to play a raw video file using the same "`mplayer`" open-source video player. This is done as the following:

```
> mplayer rawVid.yuv -demuxer rawvideo -rawvideo w=352:h=288:fps=7.5:format=i420
```

Where the characteristics of the video (e.g. width, height, frame rate, ..etc.) must be known in advance and given to the "`mplayer`" command.

* * *

A.2 PSNR CALCULATION FOR SVC VIDEOS

A.2.1 *Problem Statement*

As outlined in Chapter 2, for the objective measurement of video quality, the Peak Signal-to-Noise Ratio (PSNR) function is calculated for each frame of the video with regard to the Mean Square Error (MSE), then the average is taken over all values:

$$MSE = \frac{1}{W.H} \sum^{W} \sum^{H} [Y_r(x,y) - Y_p(x,y)]$$

$$PSNR = 10 \times \log_{10} \frac{I^2}{MSE}$$

$$PSNR_{total} = \frac{1}{N} \sum^{N} PSNR$$

where

W, H	: Frame's width and height
Y_r, Y_p	: Luminance value of the reference/processed frame in pixel (x, y)
I	: Maximum luminance value
N	: Number of frames in the video

For multistream videos, PSNR has to be calculated for each of the substreams separately. The problem is that each substream will have different values for the W, H and N parameters from the reference video, depending on the spatial and temporal resolution. The reference video has the upper bound values for those parameters. A substream of a lower spatial layer will have smaller W, H values, whereas substream of a lower temporal layer will have a smaller N value.
As the JSVM implementation provides all necessary tools to deal with SVC streams, it also includes a tool to compute PSNR values for the different layers upon encoding. However, the computed PSNR values lack precision, as shown in the result snapshot in Figure A.1.
Among 2 spatial resolutions and 5 temporal resolutions in this example, it is noticed that lower temporal layers have higher PSNR values. The inaccurate computation of PSNR that is done by JSVM will be justified in the next section.

	bitrate	Y-PSNR
176x144 @ 1.8750	93.4125	38.5063
176x144 @ 3.7500	127.9137	37.1204
176x144 @ 7.5000	165.0995	35.9945
176x144 @ 15.0000	210.6160	35.1407
352x288 @ 1.8750	404.2635	38.7535
352x288 @ 3.7500	547.1179	37.1742
352x288 @ 7.5000	703.0626	35.9920
352x288 @ 15.0000	895.7664	35.1249
352x288 @ 30.0000	1065.7728	34.4091

Figure A.1: Snapshot of PSNR Values using JSVM [2]

A.2.2 *The Limitation of PSNR Computation in JSVM*

A.2.2.1 *Difference in Temporal Resolution*

When the tested video has a lower frame rate than the reference, JSVM ignores the extra frames of the reference video. Hence, the number of performed single PSNR computations is equal to the smaller number of frames of the tested video, as shown in Figure A.2. The smaller number of PSNR computations results in a better PSNR than expected.

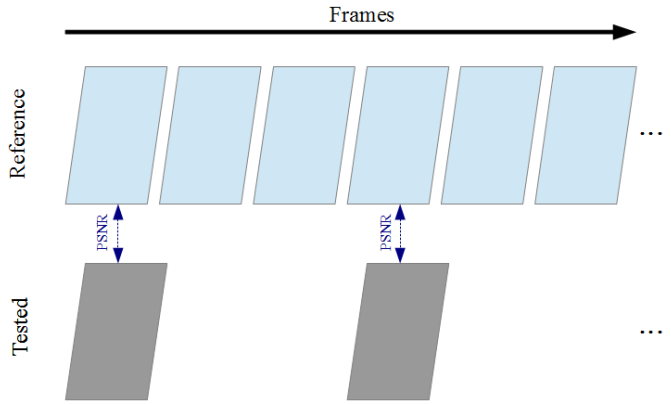


Figure A.2: The Limitation of JSVM's PSNR Computation Due to Different Temporal Resolutions

Sohn et al. [118] suggest how the computation should efficiently be performed. The video with the lower temporal resolution must duplicate its frames a number of times, so that both videos end up having the same number of frames. This is depicted in Figure A.3. Consequently, the number of single PSNR computations will be equal then to the larger number of frames, and the total PSNR value can be calculated more reliably than in the first case.

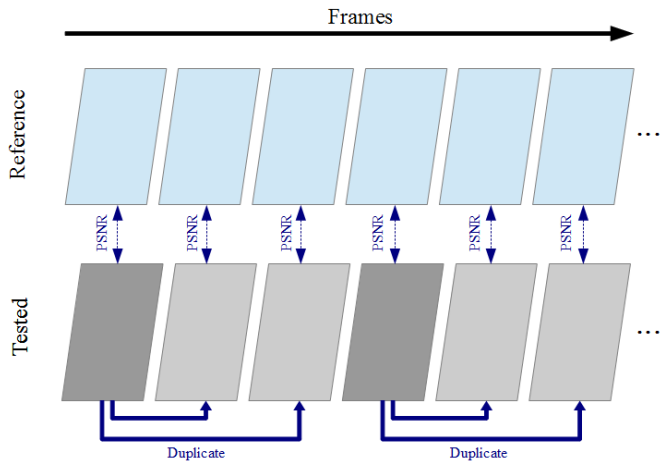
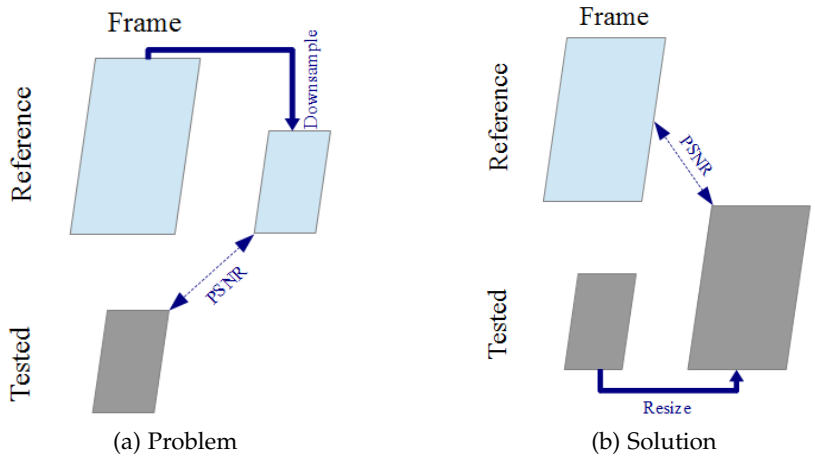


Figure A.3: Solution to Temporal Resolution Difference

A.2.2.2 *Difference in Spatial Resolution*

When the tested video has a lower spatial resolution than the reference video, JSVM downsamples the reference video to fit the smaller one, then it performs the PSNR computation in the lower resolution as shown in Figure A.4a.



(a) Problem (b) Solution

Figure A.4: The Limitation of JSVM's PSNR Computation Due to Different Spatial Resolutions: Problem & Solution

This also results in a better PSNR value than expected, because of the degradation applied to the reference video when it is downsampled. Sohn et al. [118] claim that to yield a more reliable computation, the tested video is the one that must be resized (upsampled) to the resolution of the reference video. This solution is depicted in Figure A.4b.

A.2.3 *Implementing the Proposed Solutions*

In this section, the above mentioned techniques are practically implemented. For this, we assume the presence of a raw video file "`ref.yuv`", and an SVC-encoded version of it "`encoded.264`". No matter how many layers does the SVC video contain, we will perform the PSNR computation on the Base Layer (BL), and the same can be applied on any other layer. PSNR computation is summarized by the following steps:

1. Ensure that we know the characteristics of each layer:
   `> BitStreamExtractorStatic .\vid\encoded.264`
2. Extract the current target layer (base layer in this case)
   `> BitStreamExtractorStatic .\vid\encoded.264 .\vid\L0.264 -sl 0`
3. Decode the extracted layer into the raw video format
   `> H264AVCDecoderLibTestStatic .\vid\L0.264 .\vid\x0.yuv`
4. Scale the resulted raw video temporally and spatially to match the reference video. This is done using ffmpeg, the free video manipulation tool [2]
   `> ffmpeg.exe -r 7.5 -s 176x144 -i .\vid\x0.yuv -r 30 -s 352x288 xx0.yuv`
   Where 7.5 and 176x144 are the initial values of the frame rate and spatial resolution, and 30 and 352x288 are the target values.
5. Now PSNR can be calculated between the scaled video and the reference
   `> PSNRStatic 352 288 .\vid\ref.yuv .\vid\xx0.yuv`

Where BitStreamExtractorStatic, H264AVCDecoderLibTestStatic and PSNRStatic are built-in tools in the installed JSVM library.

* * *

2 http://ffmpeg.org/index.html

A.3 FUNCTIONALITY ADAPTATION OF THE O.N.E. SIMULATOR

A.3.1 *Analysis of Adaptation Necessity*

A.3.1.1 *Problem Statement*

As presented in Section 2.2.2, Opportunistic Network Environment (ONE) is an open-source Java-based simulator, designed to implement and evaluate applications in Delay Tolerant Network (DTN) and Opportunistic Network (OppNet). The simulator can be configured with a wide of range of parameters using an external text file. Configurable parameters include for example: simulation duration, communication interface, nodes, environment, mobility model, routing protocol, messages and output reports. However, despite the powerful configurability, newly proposed functionalities can not be implemented using the textual configuration parameters. But rather, the source-code of the simulator has to be reviewed, adapted and rebuilt in order to apply core changes.

A.3.1.2 *Body Structure*

One of the few disadvantages of the ONE simulator is the lack of a detailed documentation. Therefore, a reverse-engineering analysis of the source-code had to be performed in order to analyze the structure of the simulator's body, and hence determine where the implementation of any new functionality has to go in. A short overview of the basic classes that are invoked when starting a simulation is given in the diagram in Figure A.5.

1. "`DTNSim`" is the main class that starts "`DTNSimTextUI`" in the batch mode (or "`DTNSimGUI`" in the graphical mode), which are inherited from "`DTNSimUI`"
2. "`DTNSimUI`" creates a new "`SimScenario`", which by itself is responsible for creating the hosts (nodes)
3. "`DTNSimUI`" calls *warmUp()* of "`World`", which is responsible for the movement of hosts
4. "`DTNSimTextUI`" calls *update()*, which is passed to the "`ExternalEvent`" generator
5. "`MsgCreateEvent`" passes the task of message creation to the hosts, which delegate that to the corresponding routing protocol (for example in the diagram: Spray-and-Wait (SnW))

This sequence is important to follow the execution of the simulator, and determine where the adaptation has to take place.

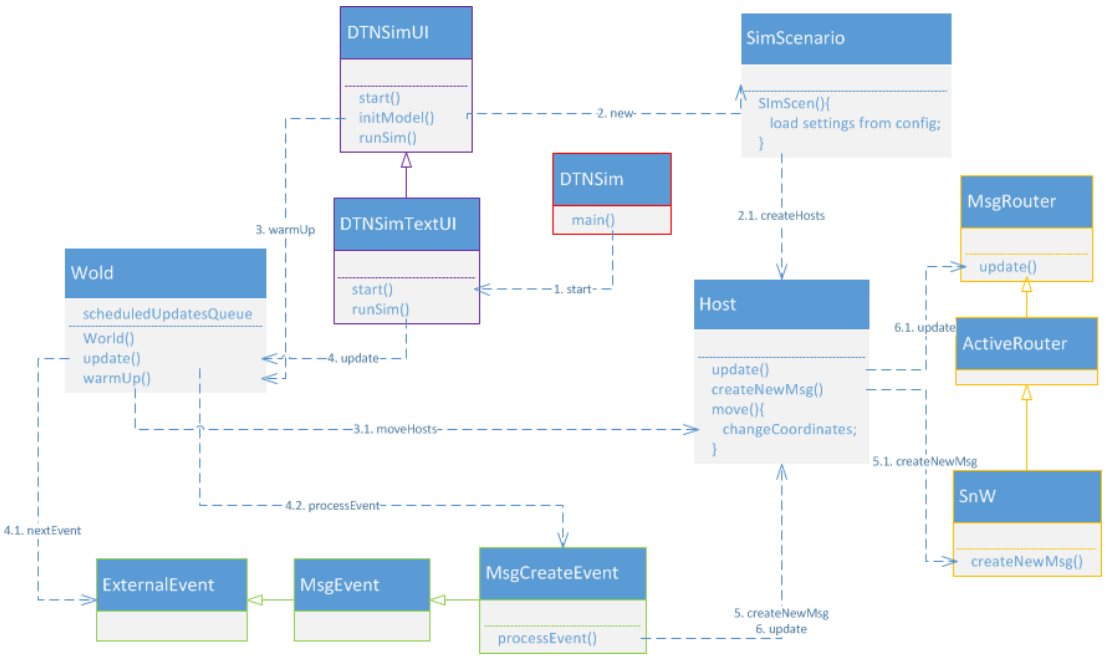


Figure A.5: ONE's Basic Classes

A.3.2 *Message Prioritizing*

For the proposal in Chapter 4, generated messages must have different redundancy factors according to their importance. The textual configuration file of ONE offers to set the redundancy factor when using SnW routing protocol as the following example.

```
SprayAndWaitRouter.nrofCopies = 100
```

However, this sets the "`number-of-copies`" property for the core of the routing protocol, and hence for all generated messages, while there is no possibility to assign this parameter for different messages differently. The responsible piece of code for the creation of messages is the function *createNewMessage()* of the "`SprayAndWaitRouter`" class, which looks like as in the Listing A.10

```
public boolean createNewMessage(Message msg) {
        makeRoomForNewMessage(msg.getSize());
        msg.setTtl(this.msgTtl);
        msg.addProperty(MSG_COUNT_PROPERTY, new Integer(initialNrofCopies));
        addToMessages(msg, true);
        return true;
}
```

Listing A.9: Message Creation in SnW

The "initialNrofCopies" parameter is read from the configuration file. Instead, we propose to ignore the value from the configuration file and overwrite the number of copies depending on the names of the messages.

```
public boolean createNewMessage(Message msg) {
        String mId = msg.getId();
        switch(mId){
                case "??": this.initialNrofCopies = ??; break;
                ...
        }
        makeRoomForNewMessage(msg.getSize());
        msg.setTtl(this.msgTtl);
        msg.addProperty(MSG_COUNT_PROPERTY, new Integer(initialNrofCopies));
        addToMessages(msg, true);
        return true;
}
```

Listing A.10: Prioritized Message Creation in SnW

Assuming that the names of the messages are predefined, the "switch" block can be filled with the corresponding redundancy factors for the different messages.

A.3.3 *Adaptive Packetization & Size Tunning*

A.3.3.1 *Update The History of Contact Times*

To keep track of the contact times history, as proposed in Chapter 5, we provided the "DTNHost" class with the following member variable:

```
public List<Double> contactTime;
```

This list has to be updated after each contact between two nodes, at each of them. To detect connections and disconnections between nodes, the "ContactTimesReport" report class implements the necessary listeners. Consequently, we add the lines of code[3] that are presented in Listing A.11.
After updating the lists at the two corresponding hosts, a host is ready to packetize its messages according to the newly updated values when coming in contact with a new host.

A.3.3.2 *Packetization*

The *packetizeMessages()* function must be implemented within the corresponding routing protocol, because it will affect the list of messages (buffer) that are ready to be

3 For simplicity, some code blocks are described as pseudo-code in "/* .. */" comments.

```
public void hostsDisconnected(DTNHost host1, DTNHost host2) {
        /* originial code */

        host1.contactTime.add(ci.getConnectionTime());
        host2.contactTime.add(ci.getConnectionTime());
}

public void hostsConnected(DTNHost host1, DTNHost host2) {
        /* originial code */

        host1.getRouter().packetizeMessages();
        host2.getRouter().packetizeMessages();
}
```

Listing A.11: Update Contact Times on Hosts Disconnections

sent. The implementation is an identical interpretation of the algorithm in Listing 5.1 in Chapter 5.

A.3.3.3 *De-Packetization*

A de-packetization procedure must take place after forwarding a message at both sender and receiver nodes, in order to extract packetized sub-messages to update their parameters, check their reception as well as prepare them for a new packetization process that is subject to the updated parameters. After a message forwarding, the sender ("`host1`") calls the function *transferDone*, and the receiver ("`host2`") calls the function *messageTransferred* from the class of the corresponding routing protocol. Both functions are adapted as in Listings A.12 and A.13.

```
protected void transferDone(Connection con) {
        String msgId = con.getMessage().getId();
        Message pM = getMessage(msgId);
        // pM is a packetizing message, loop over its sub-messages
        for(Message m : pM){
                addToMessages(m, true);
        }
        // remove pM from buffer
        removeFromMessages(pM.getId());
}
```

Listing A.12: De-Packetize at Sender

```
public Message messageTransferred(String id, DTNHost from) {
        Message pM = super.messageTransferred(id, from);
        // pM is a packetizing message, loop over its sub-messages
        for(Message m : pM){
                /*
                check if this host is the final recepient
                */
                addToMessages(m, true);
        }
        // remove pM from buffer
        removeFromMessages(pM.getId());
}
```

Listing A.13: De-Packetize at Receiver

* * *

BIBLIOGRAPHY

[1] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The one simulator for dtn protocol evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniquess*. ICST, 2009. ISBN 978-963-9799-45-5.

[2] Julien Reichel, Heiko Schwarz, and Mathias Wien. Joint scalable video model 11 (jsvm 11). *Joint Video Team, Doc. JVT-X202*, page 23, 2007.

[3] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3): 94–104, 1991.

[4] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. Mobile ad hoc networking: imperatives and challenges. *Ad hoc networks*, 1(1):13–64, 2003.

[5] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.

[6] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[7] Merza Klaghstan, David Coquil, Nadia Bennani, Harald Kosch, and Lionel Brunie. Enhancing video viewing-experience in opportunistic networks based on svc, an experimental study. In *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 3563–3567. IEEE, 2013.

[8] Merza Klaghstan, Nadia Bennani, David Coquil, Harald Kosch, and Lionel Brunie. Contact-based adaptive granularity for scalable video transmission in opportunistic networks. In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 773–778. IEEE, 2014.

[9] Merza Klaghstan, David Coquil, Nadia Bennani, Harald Kosch, and Lionel Brunie. "balcon: Backward loss concealment mechanism for scalable videos in opportunistic networks. In *2016 IEEE 27th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 1789–1795. IEEE, 2016.

[10] Scott Corson and Joseph Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. *RFC 2501*, 1999.

[11] Bhaskar Krishnamachari, Stephen B Wicker, Ramon Bejar, and Marc Pearlman. Critical density thresholds in distributed wireless networks. *Communications, information and network security*, 1:15, 2002.

[12] M Ali Rostami, Azin Azadi, and Masoumeh Seydi. Graphtea: Interactive graph self-teaching tool. In *Proc. 2014 Int. Conf. Edu. & Educat. Technol. II*, pages 48–52, 2014.

[13] Thrasyvoulos Spyropoulos, Rao Naveed Rais, Katia Turletti, Thierryand Obraczka, and Athanasios Vasilakos. Routing for disruption tolerant networks: taxonomy and design. *Wireless networks*, 16(8):2349–2370, 2010.

[14] Marco Conti and Mohan Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, 2010.

[15] Sushant Jain, Kevin Fall, and Rabin Patra. *Routing in a delay tolerant network*, volume 34. ACM, 2004.

[16] Sunil Taneja and Ashwani Kush. A survey of routing protocols in mobile ad hoc networks. *International Journal of Innovation, Management and Technology*, 1 (3):2010–0248, 2010.

[17] Laurent Franck. Performance comparison of mobile ad-hoc and delay tolerant networks. *les technologies de l'information et de la communication pour l'enseignement*, 2015.

[18] Chung-Ming Huang, Kun-chan Lan, and Chang-Zhou Tsai. A survey of opportunistic networks. In *2008.AINAW 2008. 22nd International Conference on Advanced Information Networking and Applications-Workshops*, pages 1672–1677. IEEE, 2008.

[19] Radu Ioan Ciobanu and Ciprian Dobre. Opportunistic networks: A taxonomy of data dissemination techniques. *International Journal of Virtual Communities and Social Networking(IJVCSN)*, 5(2):11–26, 2013.

[20] Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. From opportunistic networks to opportunistic computing. *Communications Magazine, IEEE*, 48(9):126–139, 2010.

[21] Samo Grasic and Anders Lindgren. An analysis of evaluation practices for dtn routing protocols. In *Proceedings of the seventh ACM international workshop on Challenged networks*, pages 57–64. ACM, 2012.

[22] Ari Keränen and Jörg Ott. Increasing reality for dtn protocol simulations. *Helsinki University of Technology, Tech. Rep*, 2007.

[23] Salman Ali, Junaid Qadir, and Adeel Baig. Routing protocols in delay tolerant networks-a survey. In *2010 6th International Conference on Emerging Technologies (ICET)*, pages 70–75. IEEE, 2010.

[24] RS Mangrulkar and Mohammad Atique. Routing protocol for delay tolerant network: A survey and comparison. In *2010 IEEE International Conference on Communication Control and Computing Technologies (ICCCCT)*, pages 210–215. IEEE, 2010.

[25] Yue Cao and Zhili Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials, IEEE*, 15(2): 654–677, 2013.

[26] Shyam Kapadia, Bhaskar Krishnamachari, and Lin Zhang. *Data delivery in delay tolerant networks: A survey*. INTECH Open Access Publisher, 2011.

[27] Evan PC Jones and Paul AS Ward. Routing strategies for delay-tolerant networks. *Submitted to ACM Computer Communication Review (CCR)*, 2006.

[28] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Single-copy routing in intermittently connected mobile networks. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 235–244. IEEE, 2004.

[29] Amin Vahdat, David Becker, et al. Epidemic routing for partially connected ad hoc networks. *Duke University Technical Report CS-200006*, 2000.

[30] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[31] Ram Ramanathan, Richard Hansen, Prithwish Basu, Regina Rosales-Hain, and Rajesh Krishnan. Prioritized epidemic routing for opportunistic networks. In *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 62–66. ACM, 2007.

[32] Xin Wang, Yantai Shu, Zhigang Jin, Qingfen Pan, and Bu Sung Lee. Adaptive randomized epidemic routing for disruption tolerant networks. In *2009 5th International Conference on Mobile Ad-hoc and Sensor Networks*, pages 424–429. IEEE, 2009.

[33] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19–20, 2003.

[34] Jérémie Leguay, Timur Friedman, and Vania Conan. Dtn routing in a mobility pattern space. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 276–283. ACM, 2005.

[35] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *World of wireless mobile and multimedia networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 183–189. IEEE, 2005.

[36] Morten Lindeberg, Stein Kristiansen, Thomas Plagemann, and Vera Goebel. Challenges and techniques for video streaming over mobile ad hoc networks. *Multimedia Systems*, 17(1):51–82, 2011.

[37] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. Manet simulation studies: the incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(4):50–61, 2005.

[38] Saba Siraj, A Gupta, and Rinku Badgujar. Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(4):199–206, 2012.

[39] Ari Keranen. Opportunistic network environment simulator. *Special Assignment report, Helsinki University of Technology, Department of Communications and Networking*, 2008.

[40] Mengjuan Liu, Yan Yang, and Zhiguang Qin. A survey of routing protocols and simulations in delay-tolerant networks. In *Wireless Algorithms, Systems, and Applications*, pages 243–253. Springer, 2011.

[41] Kevin Fall and Kannan Varadhan. The ns manual (formerly ns notes and documentation). *The VINT project*, 47, 2005.

[42] Thomas R Henderson, Sumit Roy, Sally Floyd, and George F Riley. ns-3 project goals. In *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, page 13. ACM, 2006.

[43] Andras Varga et al. The omnet++ discrete event simulation system. In *Proceedings of the European simulation multiconference (ESM 2001)*, volume 9, page 65. sn, 2001.

[44] Ari Keränen, Teemu Kärkkäinen, and Jörg Ott. Simulating mobility and dtns with the one. *Journal of Communications*, 5(2):92–105, 2010.

[45] Mirco Musolesi and Cecilia Mascolo. Mobility models for systems evaluation, a survey. *Middleware for Network Eccentric and Mobile Applications, Springer-Verlag*, 2008.

[46] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[47] Frans Ekman, Ari Keränen, Jouni Karvo, and Jörg Ott. Working day movement model. In *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40. ACM, 2008.

[48] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[49] Vivek K Goyal. Multiple description coding: compression meets the network. *IEEE Signal processing magazine*, 18(5):74–93, 2001.

[50] Fenta Adnew Mogus. Performance comparison of multiple description coding and scalable video coding. In *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pages 452–456. IEEE, 2011.

[51] Yi-Hsuan Chiang, Polly Huang, and Homer H Chen. Svc or mdc? that's the question. In *2011 9th IEEE Symposium on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, pages 76–82. IEEE, 2011.

[52] Yugang Zhou and Wai-Yip Chan. Performance comparison of layered coding and multiple description coding in packet networks. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 4. IEEE, 2005.

[53] Jacob Chakareski, Sangeun Han, and Bernd Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. *Multimedia Systems*, 10(4): 275–285, 2005.

[54] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, 17(9):1103–1120, 2007.

[55] Heiko Schwarz and Mathias Wien. The scalable video coding extension of the H.264/AVC standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, 25(2):135, 2008.

[56] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[57] Gary J Sullivan, Jill M Boyce, Ying Chen, J-R Ohm, C Andrew Segall, and Anthony Vetro. Standardized extensions of high efficiency video coding (hevc). *Selected Topics in Signal Processing, IEEE Journal of*, 7(6):1001–1016, 2013.

[58] Jill Boyce and Adarsh K. Ramasubramonian. Wd1 of shvc profiles. *Conformance Testing: http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding*, Oct. 2014.

[59] Iraide Unanue, 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. *Recent Advances on Video Coding*, 13, 2011.

[60] Peter Amon, Thomas Rathgen, and David Singer. File format for scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1174–1185, 2007.

[61] Eugene F Krause. *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation, 2012.

[62] Thomas Schierl, Cornelius Hellge, Shpend Mirta, K Gruneberg, and Thomas Wiegand. Using h. 264/avc-based scalable video coding (svc) for real time streaming in wireless ip networks. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposiumon*, pages 3455–3458. IEEE, 2007.

[63] Noel Crespi, B Molina, CE Palau, et al. Qoe aware service delivery in distributed environment. In *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*, pages 837–842. IEEE, 2011.

[64] Omneya Issa, Filippo Speranza, Tiago H Falk, et al. Quality-of-experience perception for video streaming services: Preliminary subjective and objective results. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–9. IEEE, 2012.

[65] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid–a framework for video transmission and quality evaluation. In *Computer Performance Evaluation. Modelling Techniques and Tools*, pages 255–272. Springer, 2003.

[66] Stephen Wolf and Margaret Pinson. Video quality measurement techniques. *NTIA Report 02-392*, 2002.

[67] Margaret H Pinson and Stephen Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on Broadcasting*, 50(3):312–322, 2004.

[68] Christian J Van den Branden Lambrecht and Olivier Verscheure. Perceptual quality measure using a spatiotemporal model of the human visual system. In *Electronic Imaging: Science & Technology*, pages 450–461. International Society for Optics and Photonics, 1996.

[69] Zhou Wang, Ligang Lu, and Alan C Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19 (2):121–132, 2004.

[70] Quan Huynh-Thu and Mohammed Ghanbari. The accuracy of psnr in predicting video quality for different video scenes and frame rates. *Telecommunication Systems*, 49(1):35–48, 2012.

[71] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J Karam. Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE Transactions on Broadcasting*, 57(2):165–182, 2011.

[72] Yao Yu, Yu Zhou, and Sidan Du. Video multicast over mobile ad hoc networks using network coding. In *International Conference on Management and Service Science*, pages 1–5. IEEE, 2009.

[73] Thomas Schierl, Karsten Ganger, Cornelius Hellge, Thomas Wiegand, and Thomas Stockhammer. Svc-based multisource streaming for robust video transmission in mobile ad hoc networks. *Wireless Communications, IEEE*, 13(5):96–103, 2006.

[74] Thomas Schierl, Stian Johansen, Cornelius Hellge, Thomas Stockhammer, and Thomas Wiegand. Distributed rate-distortion optimization for rateless coded scalable video in mobile ad hoc networks. In *2007 IEEE International Conference on Image Processing*, volume 6, pages VI–497. IEEE, 2007.

[75] Thomas Schierl, Thomas Stockhammer, and Thomas Wiegand. Mobile video transmission using scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1204–1217, 2007.

[76] Hulya Seferoglu and Athina Markopoulou. Opportunistic network coding for video streaming over wireless. In *Packet Video 2007*, pages 191–200. IEEE, 2007.

[77] Hulya Seferoglu and Athina Markopoulou. Video-aware opportunistic network coding over wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(5):713–728, 2009.

[78] Min Qin and Roger Zimmermann. Improving mobile ad-hoc streaming performance through adaptive layer selection with scalable video coding. In *Proceedings of the 15th international conference on Multimedia*, pages 717–726. ACM, 2007.

[79] Shiwen Mao, Shunan Lin, Shivendra S Panwar, Yao Wang, and Emre Celebi. Video transport over ad hoc networks: multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, 21(10):1721–1737, 2003.

[80] Sergio Cabrero, Xabiel G Pañeda, Thomas Plagemann, Vera Goebel, and Matti Siekkinen. Overlay solution for multimedia data over sparse manets. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 1056–1061. ACM, 2009.

[81] Christian Raffelsberger and Hermann Hellwagner. A multimedia delivery system for delay-/disruption-tolerant networks. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 530–536. IEEE, 2015.

[82] Ling-Jyh Chen, Chen-Hung Yu, Tony Sun, and Yung-Chih ChenHao-hua Chu. Effective file transfer for opportunistic networks. *Academia Sinica Technical Report TR-IIS-06-010, IIS*, 2006.

[83] Ling-Jyh Chen, Chen-Hung Yu, Cheng-Long Tseng, Hao-hua Chu, and Cheng-Fu Chou. A content-centric framework for effective data dissemination in opportunistic networks. *IEEE Journal on Selected Areas in Communications*, 26(5): 761–772, 2008.

[84] Honghai Wu and Huadong Ma. Dsvm: A buffer management strategy for video transmission in opportunistic networks. In *2013 IEEE International Conference on Communications (ICC)*, pages 2990–2994. IEEE, 2013.

[85] Hayoung Yoon, JongWon Kim, Feiselia Tan, and Robert Hsieh. On-demand video streaming in mobile opportunistic networks. In *2008. PerCom 2008. Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 80–89. IEEE, 2008.

[86] Hayoung Yoon and JongWon Kim. Peer-assisted video-on-demand in multi-channel switching wifi-based mobile networks. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–7. IEEE, 2012.

[87] Kevin Fall, Wei Hong, and Samuel Madden. Custody transfer for reliable delivery in delay tolerant networks. *IRB-TR-03-030, July*, 2003.

[88] Mikko Pitkanen, Ari Keranen, and Jörg Ott. Message fragmentation in opportunistic dtns. In *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–7. IEEE, 2008.

[89] M Renuka and P Thangaraj. Multi-path encrypted data security architecture for mobile ad hoc networks. In *2011 National Conference on Innovations in Emerging Technology (NCOIET)*, pages 153–156. IEEE, 2011.

[90] DS Vinod and G Madhusudan. Novel technique of multipath routing protocol in ad hoc network. *International Journal of Computer Networks & Communications*, 4(3):109, 2012.

[91] Yong Wang, Sushant Jain, Margaret Martonosi, and Kevin Fall. Erasure-coding based routing for opportunistic networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 229–236. ACM, 2005.

[92] Fani Tsapeli and Vassilis Tsaoussidis. Routing for opportunistic networks based on probabilistic erasure coding. In *Wired/Wireless Internet Communication*, pages 257–268. Springer, 2012.

[93] Uwe Horn, Klaus Stuhlmüller, M Link, and B Girod. Robust internet video transmission based on scalable coding and unequal error protection. *Signal Processing: Image Comm.*, 15(1):77–94, 1999.

[94] Amir Naghdinezhad, MR Hashemi, and Omid Fatemi. A novel adaptive unequal error protection method for scalable video over wireless networks. In *2007. ISCE 2007. IEEE International Symposium on Consumer Electronics*, pages 1–6. IEEE, 2007.

[95] Joohee Kim and Ji-Cheol Hong. Channel-adaptive multiple description coding for wireless video streaming. In *2007. ICCCN 2007. Proceedings of 16th International Conference on Computer Communications and Networks*, pages 474–478. IEEE, 2007.

[96] Waldir Moreira and Paulo Mendes. Survey on opportunistic routing for delay/disruptiontolerant networks. *SITI Technical Report SITI-TR-11-02*, 2011.

[97] Ling-Jyh Chen, Chen-Hung Yu, Tony Sun, Yung-Chih Chen, and Hao-hua Chu. A hybrid routing approach for opportunistic networks. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 213–220. ACM, 2006.

[98] Pierre-Ugo Tournoux, Jeremie Leguay, Farid Benbadis, John Whitbeck, Vania Conan, and Marcelo Dias de Amorim. Density-aware routing in highly dynamic dtns: The rollernet case. *IEEE Transactions on Mobile Computing*, 10(12): 1755–1768, 2011.

[99] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, volume 6, pages 1–11, 2006.

[100] Chih-Lin Hu and Bing-Jung Hsieh. A density-aware routing scheme in delay tolerant networks. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 563–568. ACM, 2012.

[101] Jani Lakkakorpi, Mikko Pitkänen, and Jörg Ott. Adaptive routing in mobile opportunistic networks. In *Proceedings of the 13th ACM international conference on Modeling,analysis, and simulation of wireless and mobile systems*, pages 101–109. ACM, 2010.

[102] Jingwei Miao, Omar Hasan, Sonia Ben Mokhtar, and Lionel Brunie. An adaptive routing algorithm for mobile delay tolerant networks. In *2011 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–5. IEEE, 2011.

[103] Yi Guo, Ying Chen, Ye-Kui Wang, Houqiang Li, Miska M Hannuksela, and Moncef Gabbouj. Error resilient coding and error concealment in scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(6):781–795, 2009.

[104] Georg Carle and Ernst W Biersack. Survey of error recovery techniques for ip-based audio-visual multicast applications. *Network, IEEE*, 11(6):24–36, 1997.

[105] Vineeth Shetty Kolkeri, MS Koul, JH Lee, and KR Rao. Error concealment techniques in h. 264/avc for wireless video transmission in mobile networks. *IEEE Fellow, Department of Electrical Engineering, University of Texas at Arlington, Advances in Engineering Science*, pages 9–16, 2008.

[106] Ying Chen, Kai Xie, Feng Zhang, Purvin Pandit, and Jill Boyce. Frame loss error concealment for svc. *Journal of Zhejiang University SCIENCE A*, 7(5):677–683, 2006.

[107] Tommi Keränen, Janne Vehkaperä, and Johannes Peltola. Error concealment for svc utilizing spatial enhancement information. In *Proceedings of the 4th International Mobile Multimedia Communications Conference (MobiMedia'08)*, 2008.

[108] Eitan Altman and Francesco De Pellegrini. Forward correction and fountain codes in delay-tolerant networks. *IEEE/ACM Transactions on Networking*, 19(1): 1–13, 2011.

[109] Mohsen Sardari, Faramarz Hendessi, and Faramarz Fekri. Dmrc: dissemination of multimedia in vehicular networks using rateless codes. In *INFOCOM Workshops 2009, IEEE*, pages 1–6. IEEE, 2009.

[110] Dieu Thanh Nguyen, Michio Hayashi, and Joern Ostermann. Adaptive error protection for scalable video coding extension of h. 264/avc. In *2008 IEEE International Conference on Multimedia and Expo*, pages 417–420. IEEE, 2008.

[111] Amir Naghdinezhad, MR Hashemi, and Omid Fatemi. A novel adaptive unequal error protection method for scalable video over wireless networks. In *2007. ISCE 2007. IEEE International Symposium on Consumer Electronics*, pages 1–6. IEEE, 2007.

[112] Selim Aksoy and Robert M Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*, 22(5): 563–582, 2001.

[113] Munchurl Kim, Sangjin Hahm, Keunsik Lee, Keunsoo Park, et al. A partial protection scheme based on layer dependency of scalable video coding. In *2008. VIE 2008. 5th International Conference on Visual Information Engineering*, pages 777–782. IET, 2008.

[114] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90, 2008.

[115] M Reisslein. Video trace library. *Arizona State University,[online] Available: http://trace. eas. asu. edu*, 2012.

[116] Ye-Kui Wang, Stephan Wenger, and Miska M Hannuksela. Common conditions for svc error resilience testing. *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q*, 6: 7–10, 2005.

[117] J. Chen, D. Wu, H. Gao, and J.Y. Tham. A novel scheme of generating streamable mp4 files for scalable video coding. In *2009. AH-ICI 2009. First Asian Himalayas International Conference on Internet*, pages 1–5. IEEE, 2009.

[118] Hosik Sohn, Hana Yoo, Wesley De Neve, Cheon Seog Kim, and Yong Man Ro. Full-reference video quality metric for fully scalable and mobile svc content. *IEEE Transactions on Broadcasting*, 56(3):269–280, 2010.

[119] Arno Barzan, Bram Bonné, Peter Quax, Wim Lamotte, Mathias Versichele, and Nico Van de Weghe. A comparative simulation of opportunistic routing protocols using realistic mobility data obtained from mass events. In *2013 IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, 2013.

[120] Philip Ginzboorg, Valtteri Niemi, and Jörg Ott. Message fragmentation for disrupted links. In *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*, pages 1–10. VDE, 2011.

[121] Wei Gao and Guohong Cao. On exploiting transient contact patterns for data forwarding in delay tolerant networks. In *2010 18th IEEE International Conference on Network Protocols (ICNP)*, pages 193–202. IEEE, 2010.

[122] Friedrich Pukelsheim. The three sigma rule. *The American Statistician*, 48(2): 88–91, 1994.

[123] Marco Brandas, Maria G Martini, Mikko Uitto, and Janne Vehkapera. Quality assessment and error concealment for svc transmission over unreliable channels. In *2011 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2011.

[124] Médéric Blestel and Mickaël Raulet. Open svc decoder: a flexible svc library. In *Proceedings of the international conference on Multimedia*, pages 1463–1466. ACM, 2010.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : Klaghstan
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 01/12/2016

Prénoms : Merza

TITRE : Multimedia Data Dissemination in Opportunistic Systems

NATURE : Doctorat

Numéro d'ordre : 2016LYSEI125

Ecole doctorale : N° ED512 : Informatique et Mathématiques de Lyon

Spécialité : Informatique

RESUME :

Les réseaux opportunistes se forment spontanément et dynamiquement grâce à un ensemble d'utilisateurs itinérants dont le nombre et le déplacement sont imprévisibles. La diffusion de bout-en-bout d'informations, dans ce contexte, est incertaine du fait de la forte instabilité des liens réseaux entre les utilisateurs. Alors que la plupart des travaux qui en ont envisagé l'usage visent la transmission de messages de faible taille, la transmission de données plus volumineuses telles que les vidéos représente une alternative très pertinente aux réseaux d'infrastructure.
La diffusion de vidéos dans des réseaux opportunistes constitue cependant un challenge important. En effet, permettre, dans ce contexte, au destinataire d'une vidéo de prendre connaissance au plus vite du contenu de celle-ci, avec la meilleure qualité de lecture possible et en encombrant le moins possible le réseau reste un problème largement ouvert.
Cette thèse propose une solution à cette probématique. Celle-ci est basée sur le choix de l'encodage SVC subdivisant la vidéo en un ensemble de couches interdépendantes (layers), chacune améliorant la précédente en terme de résolution, de densité, ou de perception visuelle.
Notre solution se décline en trois contributions. La première est une adaptation du mécanisme de diffusion Spray-and-Wait aux spécificités de l'encodage SVC. Les couches sont ainsi diffusées avec un niveau de redondance propre à chacune, adapté à leur degré d'importance dans la diffusion de la vidéo.
Notre seconde contribution consiste à améliorer, sans surcoût, le mécanisme précédent en prenant en compte une granularité plus fine (NLU) et adaptative en fonction de l'évolution de la topologie du réseau.
Enfin, la troisième contribution consiste à proposer un mécanisme hybride de complétion des couches vidéos arrivées incomplètes à destination. Cette méthode, initiée par le destinataire, combine un protocole de recherche sur le réseau des NALUs manquants et des techniques de complétion à base d'opérations sur les frames constituant la vidéo.
La faisabilité de nos contributions et l'étude de l'influence des différentes caractéristiques ont pu être testés grâce à l'implantation de nos algorithmes sur l'environnement de simulation des oppnets (ONE). Les résultats obtenus sont prometteurs et ouvrent de nouvelles perspectives de recherche dans le domaine.

MOTS-CLÉS : réseaux opportunistes, Protocoles distribués, diffusion de vidéos

Laboratoire (s) de recherche : LIRIS

Directeur de thèse: Prof. Lionel Brunie

Président de jury : Prof. Michael Granitzer

Composition du jury :

Prof. Jaime Delgado,	*Universitat Politècnica de Catalunya, Spain*,	Rapporteur
Assoc.-Prof. Ahmed Mostefaoui,	*Université de Franche-Comté, France*,	Rapporteur
Assoc.-Prof. Klaus Schöffmann,	*Universität Klagenfurt, Austria*,	Examinateur
Prof. Michael Granitzer,	*Universität Passau, Germany*,	Examinateur
Assoc.-Prof. Faiza Najjar,	*Ecole nationale des sciences* de l'Informatique, Tunisia,	Examinatrice
Prof. Lionel Brunie,	*INSA de Lyon, France*,	Directeur de thèse
Prof. Harald Kosch,	*Universität Passau, Germany*,	Directeur de thèse
Assoc.-Prof. Nadia Bennani,	*INSA de Lyon, France*,	Co-directeur de thèse