Mobile data offloading via urban public transportation networks

Qiankun Su

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :

M. QIANKUN SU

le vendredi 19 mai 2017

Titre :

Mobile Data Offloading via Urban Public Transportation Networks

---

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. CHARLY POULLIAT

MME KATIA JAFFRES RUNSER

Rapporteurs :

M. MARCO FIORE, CONSIGLIO NAZIONALE DE RECERCHE

M. YACINE GHAMRI-DOUDANE, UNIVERSITE DE LA ROCHELLE

Membre(s) du jury :

Mme ISABELLE GUERIN LASSOUS, UNIVERSITE LYON 1, Président

M. CHARLY POULLIAT, INP TOULOUSE, Membre

M. GENTIAN JAKLLARI, INP TOULOUSE, Membre

Mme KATIA JAFFRES RUNSER, INP TOULOUSE, Membre

Acknowledgements

First of all, I would like to express my special appreciation and thanks to my advisors, Prof. Katia Jaffrès-Runser, Prof. Gentian Jakllari and Prof. Charly Poulliat, who gave me all the support and encouragement for needed to realize this thesis throughout my Ph.D. study. Besides, I owe my gratitude to the rest of my thesis committee, Prof. Marco Fiore, Prof. Yacine Ghamri-Doudane and Prof. Isabelle Guérin-Lassous, for their insightful comments and suggestions.

Secondly, I would like to acknowledge my sponsor, the joint program between the University of Toulouse and the China Scholarship Council (CSC), for giving me access to the laboratory, and research facilities and providing me with the financial security. Thanks to this, I can focus purely on my studies and research.

Finally, I am also thankful to my family for their understanding and mental support. Also, I would like to thank my friends who gave me advice in writing this thesis and encouraged me striving towards my goal.

Abstract

Mobile data traffic is increasing at an exponential rate with the proliferation of mobile devices and easy access to large contents such as video. Traffic demand is expected to soar in the next 5 years and a new generation of mobile networks (5G) is currently being developed to address the looming bandwidth crunch. However, significant 5G deployments are not expected until 2020 or even beyond. As such, any solution that offloads cellular traffic to other available networks is of high interest, the main example being the successful offloading of cellular traffic onto WiFi.

In this context, we propose to leverage public transportation networks (PTNs) created by regular bus lines in urban centers to create another offloading option for delay tolerant data such as video on demand. This PhD proposes a novel content delivery infrastructure where wireless access points (APs) are installed on both bus stops and buses. Buses act as data mules, creating a delay tolerant network capable of carrying content users can access while commuting using public transportation. Building such a network raises several core challenges such as: (*i*) selecting the bus stops on which it is best to install APs, (*ii*) efficiently routing the data, (*iii*) relieving congestion points in major hubs and (*iv*) minimizing the cost of the full architecture. These challenges are addressed in the three parts of this thesis.

The first part of the thesis presents our content delivery infrastructure whose primary aim is to carry large volumes of data. We show that it is beneficial to install APs at the end stations of bus lines by analyzing the publicly available time tables of PTN providers of different cities. Knowing the underlying topology and schedule of PTNs, we propose to

pre-calculate static routes between stations. This leads to a dramatic decrease in message replications and transfers compared to the state-of-the-art Epidemic delay tolerant protocol. Simulation results for three cities demonstrate that our routing policy increases by 4 to 8 times the number of delivered messages while reducing the overhead ratio.

The second part of the thesis addresses the problem of relieving congestion at stations where several bus lines converge and have to exchange data through the AP. The solution proposed leverages XOR network coding where encoding and decoding are performed hop-by-hop for flows crossing at an AP. We conduct a theoretical analysis of the delivery probability and overhead ratio for a general setting. This analysis indicates that the maximum delivery probability is increased by 50% while the overhead ratio is reduced by 50%, if such network coding is applied. Simulations of this general setting corroborate these points, showing, in addition, that the average delay is reduced as well. Introducing our XOR network coding to our content delivery infrastructure using real bus timetables, we demonstrate a 35% - 48% improvement in the number of messages delivered.

The third part of the thesis proposes a cost-effective architecture. It classifies PTN bus stops into three categories, each equipped with different types of wireless APs, allowing for a fine-grained cost control. Simulation results demonstrate the viability of our design choices. In particular, the 3-Tier architecture is shown to guarantee end-to-end connectivity and reduce the deployment cost by a factor of 3 while delivering 30% more packets than a baseline architecture. It can offload a large amount of mobile data, as for instance 4.7 terabytes within 12 hours in the Paris topology.

Résumé

La popularité des plateformes mobiles telles que smartphones et tablettes génère un volume croissant de données à transférer. La principale raison de cette croissance est l'accès simplifié aux contenus vidéo sur ces plateformes. La future génération (5G) de téléphonie mobile est en cours de développement et a pour objectif d'offrir une bande passante suffisante pour de tels volumes de données. Néanmoins, un déploiement en masse de la 5G n'est pas envisagé avant 2020. De plus, la croissance est telle qu'il sera forcément intéressant de développer des solutions alternatives et complémentaires capables de délester le réseau cellulaire. L'exemple actuel le plus représentatif est le délestage de données cellulaires vers des réseaux d'accès WiFi par les principaux opérateurs mobiles.

Dans ce contexte, nous proposons de déployer un nouveau réseau de contenus qui s'appuie sur les réseaux de transport publics urbains. Cette solution déploie des bornes sans-fil dans les bus et sur certaines stations de bus pour offrir du contenu aux passagers des bus. Les bus enregistrent et transportent les données, et se comportent donc comme des mules qui peuvent s'échanger des données dans certaines stations de bus. L'ensemble des bus créé un réseau de transport de données tolérantes au délai telles que de la vidéo à la demande. La création d'un tel réseau soulève de nombreuses questions. Les questions traitées dans les trois parties de cette thèse sont les suivantes: (i) le choix des stations de bus sur lesquelles une borne sans-fil doit être déployée, (ii) le choix du protocole de routage des données, (iii) la gestion efficace de la contention dans les stations et enfin (iv) la réduction du coût d'une telle infrastructure.

La première partie de la thèse présente notre réseau de contenu dont l'objetif principal est de transporter de larges volumes de données. Nous montrons pour cela qu'il suffit de déployer des bornes sans-fil aux terminus des lignes de bus. Ce résultat provient de l'analyse des réseaux de transports publics des villes de Toulouse, Helsinki et Paris. Connaissant les horaires et la topologie de ces réseaux de transport, nous proposons de pré-calculer les routes pour transmettre les données dans ce réseau. Nous montrons que ce routage statique permet de réduire drastiquement le nombre de réplications de messages quand on le compare à un routage épidémique.

La seconde contribution de cette thèse s'intéresse à l'échange des messages au niveau des bornes sans-fil déployées aux terminus des lignes de bus. En effet, les protocoles d'accès actuels partagent équitablement la bande passante entre les bus et le point d'accès. Dans notre cas, il en résulte une congestion importante que nous proposons de résoudre en introduisant un codage réseau XOR de proche en proche. Les flux qui se croisent sont alors combinés par la borne. Les bus transportent des paquets codés qui seront décodés au prochain saut par la borne suivante. Une analyse théorique de ce mode de communication montre que la probabilité de reception des messages peut-être augmentée au maximum de 50% et la surcharge diminuée au maximum de 50%. Pour les 3 villes Européennees considérées, nous montrons par simulation que ce protocole permet d'augmenter de 35% à 48% le nombre de messages reçus.

La dernière partie de cette thèse a pour objectif de réduire le coût de déploiement d'une telle architecture. Elle classifie les terminus des lignes de bus en trois ensembles qui sont équipés par des bornes sans fil de nature différentes. Les résultats de simulation montrent que pour les trois villes il est possible de guarantir la connectivité de bout-en-bout tout en réduisant les coûts de déploiement d'un facteur 3. Cette architecture, dénommée 3-tier, transporte 30% plus de messages que

le déploiement initial proposé en première partie. Pour conclure, nous montrons qu'il est possible de décharger un grand volume de données avec notre architecture. Par exemple, pour Paris, notre architecture permet de délester jusqu'à 4.7 téraoctets de données en 12 heures de temps.

Contents

List of Figures

List of Tables

Acronyms

ICT	Information Communications Technology
DTN	Delay (or Disruption) Tolerant Network
VANET	Vehicular Ad-Hoc Network
PTN	Public Transportation Network
MAC	Medium Access Control
AP	Access Point
BS	Base Station
NC	Network Coding
RLNC	Random Linear Network Coding
CDS	Connected Dominating Set
MCDS	Minimum Connected Dominating Set
BW	Betweenness Centrality
PR	PageRank
ONE	Opportunistic Network Environment
GTFS	General Transit Feed Specification

Introduction

The world has experienced tremendous urban growth in recent decades with 70% of the world's population expected to live in urban areas by 2050 [2]. Cities are responding to the environmental, transportation and infrastructure challenges this raises by becoming more intelligent, interconnected and efficient, making information and communications technologies (ICTs) central to their success [3]. At the same time, the emergence of the smartphone as the main platform for Internet access has placed a big strain on the ICT infrastructure in urban areas. Globally, mobile data traffic has grown 4,000-fold over the past 10 years and will increase nearly 8-fold, at an annual growth rate of 53% between 2015 and 2020, reaching 30.6 exabytes per month by 2020 [4].

The explosive growth of demand for mobile capacity over the past decades has pushed researchers and industries to upgrade cellular networks, from the early analog telecommunications standards (1G) in the 1980s to the current mature mobile wireless technologies (4G) [5]. However, 4G is insufficient to meet the rapid demands of the mobile telecommunication markets since only incremental improvements and small amounts of new spectrum can be expected going forward[6]. A new generation mobile network (5G) is envisioned to address the looming bandwidth crunch. However, significant 5G deployments are not expected until 2020 or beyond due to still unresolved regulatory, spectrum availability and new infrastructure deployment challenges [7].

In this thesis, we propose a novel content delivery infrastructure that relies on off-the-shelf technology and the public transportation network (PTN) to help relieve the bandwidth crunch in urban areas. Our proposition is to install WiFi access points at selected bus stations, with only a subset of them having Internet

access, and buses, using the latter as data mules. This will create a data-mule delay tolerant network capable of carrying content PTN customers can access while on the bus or waiting at selected stations. Data is updated onboard buses when they connect to wireless access points deployed at selected bus stops in the network. Similarly, data stored on buses can be pushed to the bus stop access points to be routed or disseminated further in the network. Mobile users connect to the platform on the bus to download different kinds of content, including videos, books, news, etc., and can as well publish new content.

The main advantage of the proposed content delivery infrastructure is its low cost since it relies on inexpensive WiFi technology and an extensive public transportation network already in place. Also, such a network has low packet loss as contents are transmitted while public vehicles park at stations rather than move. Thanks to the long contact duration between buses and stations, transferring large contents, like videos, becomes possible.

Throughout this work, we gradually improve the network performance of our content delivery infrastructure in terms of packet delivery and cost effectiveness by leveraging the structural properties of the underlying bus network and efficient network coding strategies.

Motivation

Growing urbanization leads to an increasing demand for public transport. A significant part of the mobile content is consumed in urban areas while people are commuting using public transportation. To deal with the explosive growth in mobile data traffic in urban areas, vehicular networks are used to offload data away from cellular networks. The most well-known example of such networks is Vehicular Ad-Hoc Networks (VANETs). However, they are designed to create networks for delay-sensitive communications and are mainly targeted for private transports. A solution for massive offloading using private electrical vehicles has been investigated in [8]. This solution works for long distance (country-wide) transport of bulk data. Data is carried by electrical vehicles and is exchanged at charging stations. Results are promising as 20% of vehicles in circulation in France equipped with only one Terabyte of storage can offload Petabyte transfers

in a week. The solution we propose in this manuscript is complementary and targets urban-scale offloading of delay-tolerant data.

Some works have been done to leverage public transportation networks to create a wireless infrastructure. For instance, DakNet [9] provides low-cost digital communication for isolated and non-connected villages in India and Cambodia by using public buses as data mules. Inspired by these works, in this thesis, we propose a novel content delivery infrastructure leveraging the existing public transportation networks (PTNs) of dense cities. WiFi access points are deployed on selected public bus stations and buses, using the latter as data mules, creating a delay tolerant network capable of carrying content that users can access while using the public transportation.

Our content delivery infrastructure is prone to congestion points as public transportation networks are built around the concept of hubs where many bus lines meet and exchange data. Legacy wireless medium access control protocols, including IEEE 802.11, are designed to share the channel fairly among radios. Traffic exchanges between buses and station are unbalanced, creating strong congestion points. To relieve congestion points, we propose to use network coding, which has been shown to significantly improve the system throughput in such a situation.

Installing network coding enabled APs at all PTN end stations is expensive as it requires specialized hardware plus extended storage capabilities. In addition, network coding suffers from a new security thread, namely, pollution attacks. The attack spreads the pollution by combining legitimate messages with polluted ones and therefore limits the recovery probability of legitimate messages [10]. To address the challenge arising from constructing a cost-effective platform, we introduce a 3-Tier architecture that classifies a number V of PTN bus stops into three categories, each equipped with a different type of AP. To realize this architecture we address several challenges. First, we identify the minimum number of bus stops, V', for guaranteeing end-to-end connectivity by using connected dominating sets. Second, among the bus stops, we select a very small subset V'', considered to be the most important, as the first tier. As such, the 3-tier architecture is composed of a first tier equal to V'' where network coding-enabled wireless AP is deployed, a second tier equal to $V' - V''$ where regular wireless AP is deployed, and a third tier equal

to $V - V'$ where no AP is deployed.

Outline

This thesis is organized as follows.

In Chapter 1, we introduce the general context and the main related works of my research on delay tolerant networks that leverage public transportation networks, including routing protocols, network coding.

In Chapter 2, we give the details of the proposed content delivery infrastructure and highlight the routing protocol we recommend. We evaluate the performance of basic roll-out using real PTN schedules for Toulouse, Paris and Helsinki.

In Chapter 3, we introduce network coding within our infrastructure. We derive for various scenarios a closed-form expression for the potential gain of network coding. We verify the theoretical analysis through intensive simulations.

In Chapter 4, we introduce a cost-effective architecture that guarantees end-to-end delivery and minimizes hardware cost by categorizing stations into three tiers, each of which is deployed with different types of wireless access points.

In Chapter 5, we conclude this work and also outline directions for future research.

Contributions

The main contributions of this thesis can be summarized as follows:

- A Content Delivery Infrastructure Leveraging Urban Public Transportation Networks [11] [1]
  - We propose a novel content delivery infrastructure that can relieve the wireless bandwidth crunch in dense urban centers.
  - Based on our analysis of real traces of PTNs, we only deploy wireless access points at end stations rather than all bus stops which is way too much.
  - We adapt to the underlying network topology of PTNs and messages is delivered following the shortest path, which avoids flooding unnecessary

message copies. Trace-based simulations show 4 to 8 fold improvement in delivered messages compared with a legacy routing protocol.

- XOR Network Coding for the Content Delivery Infrastructure [11] [1] [12] [2]
  - We leverage inter-session XOR hop-by-hop network coding to address the congestion points in major hubs. It has the advantage of high coding probability, low message delay and simple replication control.
  - A theoretical analysis is carried out that indicates that the maximum delivery probability is increased by 50% and the maximum overhead ratio is reduced by 50% if our network coding strategy is applied. Our simulations verify these points, showing the average delay is reduced as well.
  - We design encoding and decoding procedures for our content delivery infrastructure. Trace-based simulations show our network coding scheme improves by 35%-48% in delivered messages.
- Towards a Cost-effective Design [11] [1]
  - We propose a cost-effective architecture to provide the end-to-end connectivity, high packet delivery and minimize hardware cost.
  - The proposed architecture achieves a factor of 3 reductions in the number of access points required while delivering 30% more messages than a baseline architecture.
  - Trace-based simulations also show that the architecture can offload up to a large amount of data, as for instance 4.7 terabytes within 12 hours in the Paris topology.

1. Qiankun Su, Katia Jaffrès-Runser, Gentian Jakllari and Charly Poulliat, "**An efficient content delivery infrastructure leveraging the public transportation network**," in *19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'16)*, Malta, Nov. 2016. pp. 338-347.
2. Qiankun Su, Katia Jaffrès-Runser, Gentian Jakllari and Charly Poulliat, "**Xor network coding for data mule delay tolerant networks**", in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, 2015, pp. 1–6. (invited paper)

1 State of the Art

This chapter introduces the context and main concepts relevant to the study of delivering contents using a public transportation network. As our infrastructure constitutes a delay tolerant network (DTN), we focus on major works related to DTN with a special emphasis on DTNs where node's trajectories are predictable. We also introduce a list of related works on network coding in DTNs that we use to address congestion points arisen from our content delivery infrastructure.

1.1 Urban wireless infrastructures

Mobile data traffic is growing fast with the proliferation of mobile devices, in particular smartphones. According to the report from the Online Publishers Association (OPA) and research firm Frank N. Magid Associates, the Internet usage on mobile devices has exceeded PC usage since early 2014 [13]. Globally, mobile data traffic has grown 4,000-fold over the past 10 years and will increase nearly 8-fold at an annual growth rate of 53% between 2015 and 2020, reaching 30.6 exabytes per month by 2020 [4]. In past decades, a diversity of wireless communication technologies have been applied to different application scenarios, including cellular networking, wireless local area networking, personal networking or satellite communication. The best-known example of mobile wireless digital communication systems is cellular networks [14].

1.1.1 Cellular networks

A cellular network is a communication network distributed over land areas though cells where each cell contains at least a fixed location transceiver known as

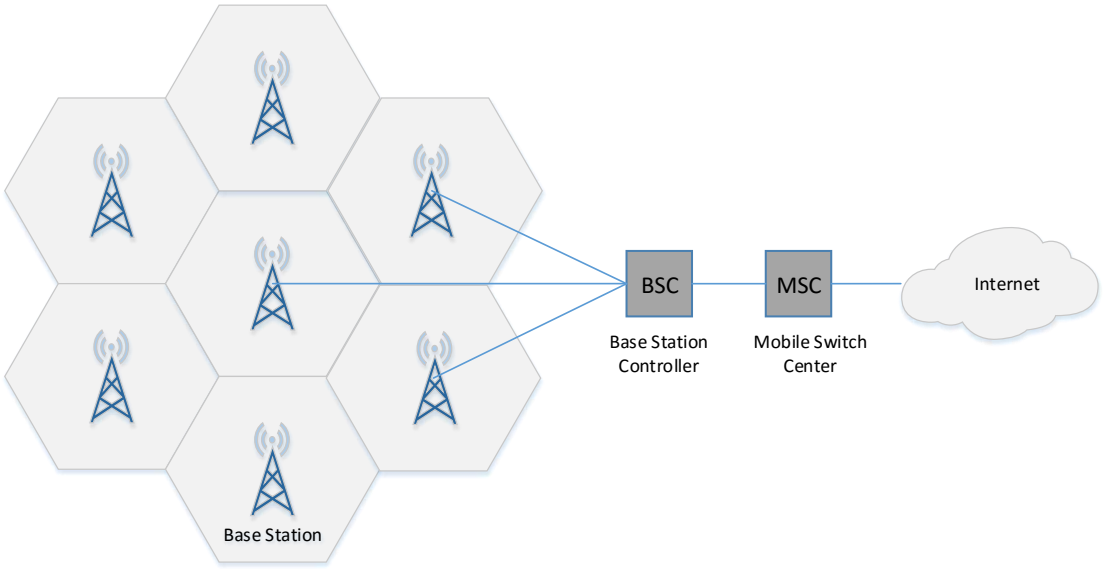


Figure 1.1 – Cellular networks

Base Station (BS), as depicted in Figure 1.1. These cells together provide radio coverage over city-wide/national/global areas. It allows user equipment, such as mobile phones, to move seamlessly even during transmission.

The explosive growth of demand for mobile capacity over the past decades has pushed researchers and industries to upgrade cellular networks, from the early analog telecommunications standards (1G) in 1980 to the current mature mobile wireless technologies (4G) in 2010 [5]. However, 4G is not sufficient to meet the rapid demands of the mobile telecommunication market since only incremental improvements and small amounts of new spectrums can be expected on it [6]. A new generation mobile networks (5G) is currently developed to address the looming bandwidth crunch. However, significant 5G deployments are not expected until 2020 or beyond due to still unresolved regulatory, spectrum availability and new infrastructure deployment challenges [7].

1.1.2 Mobile data offloading

The above-mentioned circumstances fostered the interest in the use of complementary network technologies and innovative techniques to mitigate the burden on cellular networks, which is so-called **mobile data offloading** or simply data

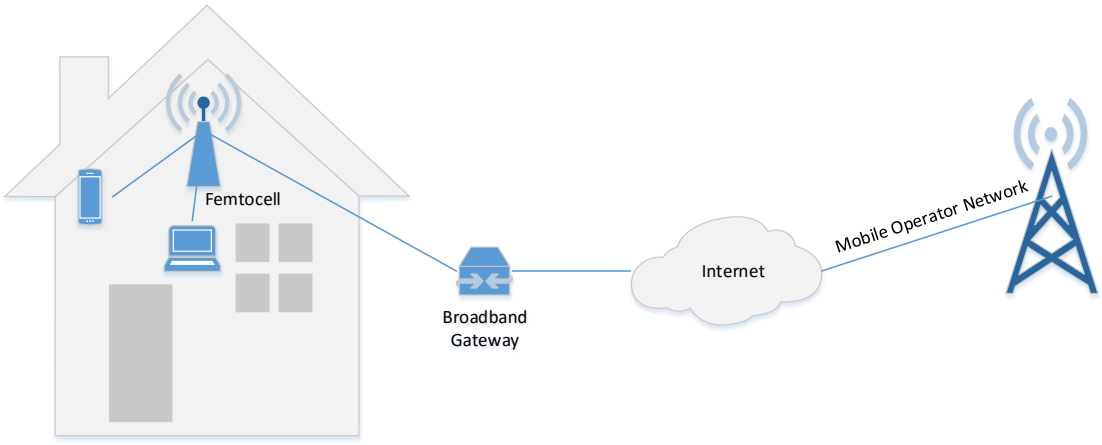


Figure 1.2 – Femtocell networks

offloading [15]. It can reduce the amount of data being carried on the cellular bands and free bandwidth for other users. Femtocell and Wi-Fi technologies are two widely used mobile data offload technologies. According to the white paper of Cisco Visual Networking Index (VNI) [4], offloaded mobile traffic exceeded cellular traffic for the first time in 2015. Specifically, 51% of total mobile data traffic was offloaded onto the fixed network through femtocell or WiFi in 2015.

1.1.3 Data offloading via femtocells

Femtocell networks or home base stations are such networks where consumers install femto access points (a small, low-power cellular base station) inside their homes and backhaul data goes through a broadband gateway over the Internet to the cellular operator network [16, 17], as illustrated in Figure 1.2.

Femtocells are attractive to both mobile operators and consumers as they provide improvements in both capacity and coverage, especially indoors [17, 18]. The capacity is improved by a reduction in data being carried on the cellular bands and the freed capacity enhances the experience of customers on cellular networks. The coverage is also improved as femtocells eliminate signal attenuation during the transmission of cellular bands though buildings. As such, indoor users have a much better user experience with higher data rates. In addition, user equipment prolongs their battery life due to the reduced trasmitter–receiver distance.

However, femtocells operate in costly licensed and limited spectrum bands as macrocells (the widest range of cell sizes), which may lead to interference during data transmission [19]. Instead, WiFi technology allows data traffic to be shifted from expensive licensed bands to free unlicensed bands (2.4 GHz and 5 GHz) [18]. As WiFi works on the different frequency as cellular bands, it has no impact on existing cellular networks.

1.1.4 Data offloading via WiFi

WiFi is a wireless connectivity solution based on IEEE 802.11 standards that allows electronic devices to connect to a wireless local area network (WLAN). It has emerged as a promising and cost-effective solution to offload data traffic away from congested cellular networks for the following main reasons [20, 21, 22, 23]: *i*) Most Internet-capable mobile devices (e.g., smartphones, tablets, laptops) have the built-in WiFi capabilities; *ii*) No licensed spectrum is required. WiFi devices operate on free world-unified 2.4 GHz and 5 GHz bands; *iii*) WiFi has high data rates. Data rates up to 600 Mbps and 6.9 Gbps is achieved by IEEE 802.11n and 802.11ac respectively.

Femtocells and WiFi are complementary to each other. Femtocells provide a seamless service to users with low data rates while WiFi offers high data rates but QoS (Quality of Service) unguaranteed [18]. However, apparently, the mobility of both femtocell and WiFi networks is constrained on indoors (e.g., home, office). Once leaving the indoors, users access the Internet via cellular networks again.

Some studies have shown that it is possible to offload data while users are outdoors. In the following section, we present the main data offloading technologies that can handle the mobility of outdoor users and are candidates for providing larger bandwidth to future communications.

1.2 Data offloading over vehicular networks

Over the last few years, we have witnessed the rapid growth of vehicles in urban areas together with the increase of internet-enabled devices integrated into the vehicles [24, 25]. The data created by these devices worsens congested cellu-

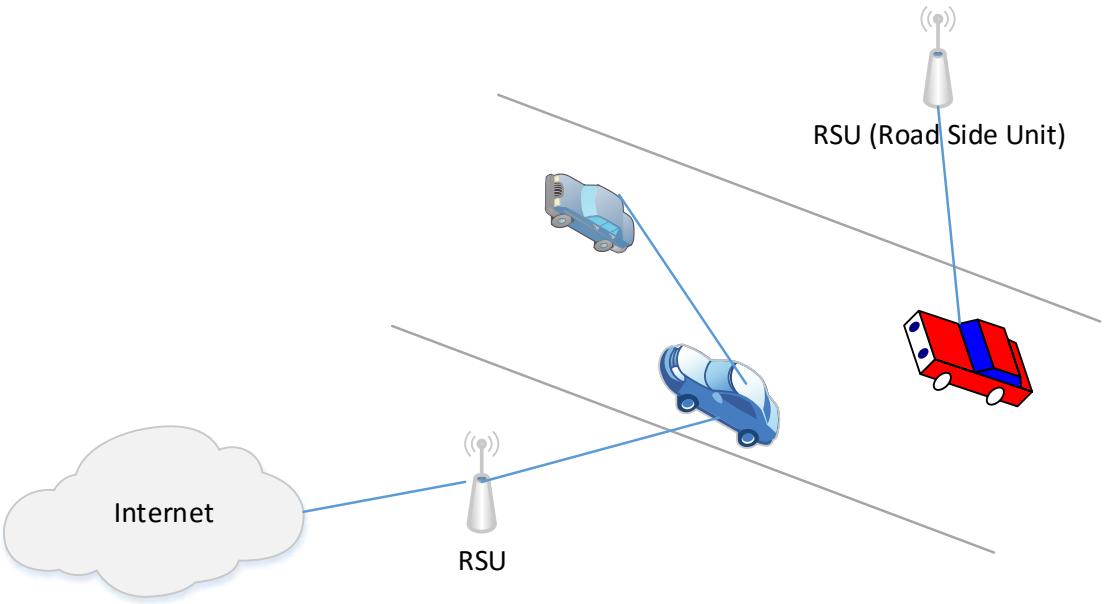


Figure 1.3 – VANETs

lar networks. Several studies have been carried out using vehicular networks to shift bandwidth-hungry traffic from cellular networks. The most popular of such networks is Vehicular Ad-Hoc Network (VANET).

1.2.1 Offloading over Vehicular Ad-Hoc Networks

Vehicular Ad-Hoc Networks (VANETs) use cars as mobile nodes to create a mobile ad-hoc network. Rather than moving at random, vehicles tend to move in predefined road paths. It provides a wireless communication for *private* transports [26] where vehicles can communicate with either another vehicle or fixed equipment situated next to the road (that is so-called RSU, Road Side Unit), as shown in Figure 1.3.

A wide range of delay-sensitive applications can be explored in VANETs, including automatic collision warning, remote vehicle diagnostics, emergency management and assistance for safe driving [24, 25]. VANETs are also used to offload data away from cellular networks [27][8]. As depicted in Figure 1.3, vehicles can download or upload contents (such as images, audio and video) through RSU rather than cellular bands when they pass through.

However, as vehicles in VANETs move in high velocities during data transmission, it may lead to link failures and packet losses. The packet loss rates are around 20% as cars move [28]. Also, transmitting big files, such as audios and videos, becomes very challenging because of short contact durations (typically within a few seconds) among vehicles or between the vehicle and the road side units.

Unlike VANETs mainly targeted for cars, we explore in this Ph.D. work a solution to offload a large volume of data while passengers use public transportation networks (PTNs). The goal of this work is to leverage PTNs to convey delay tolerant data. Our aim is different from the one of [8] where large data has to be transferred over very long distances. The data we consider in this work will be consumed by bus passengers while this is not the case in [8].

1.2.2 Offloading over public transportation networks

Growing urbanization leads to an increasing deployment of public transports. Currently, passengers onboard access to the Internet via cellular networks directly or indirectly. Main business solutions are to install a robust industrial 3G/4G router which connects to cellular networks, as illustrated in Figure 1.4. The service works like any fixed hotspot in a hotel, coffee shop or airport, providing customers an onboard Wi-Fi hotspot service to check email, surf the web and local entertainment during their journey. It is currently available mostly for some long-distance coaches but rarely in urban public transportation networks (PTNs). These solutions are heavily dependent on cellular networks and will bring lots of stress to the existing urban cellular infrastructure.

Some researchers explored the idea of leveraging public transportation networks to create a wireless infrastructure. Real implementations of such networks have been tested in the last decade.

DakNet [9, 29], developed by MIT Media Lab researchers, provides low-cost digital communication for remote villages in India and Cambodia by using public buses as data mules. In this case, a public bus carries a mobile access point (MAP) between remote villages and a hub with Internet access. Data automatically uploads and downloads when the bus is in the range of the village or the hub. Due to inadequate infrastructures in remote villages, access points are deployed at all

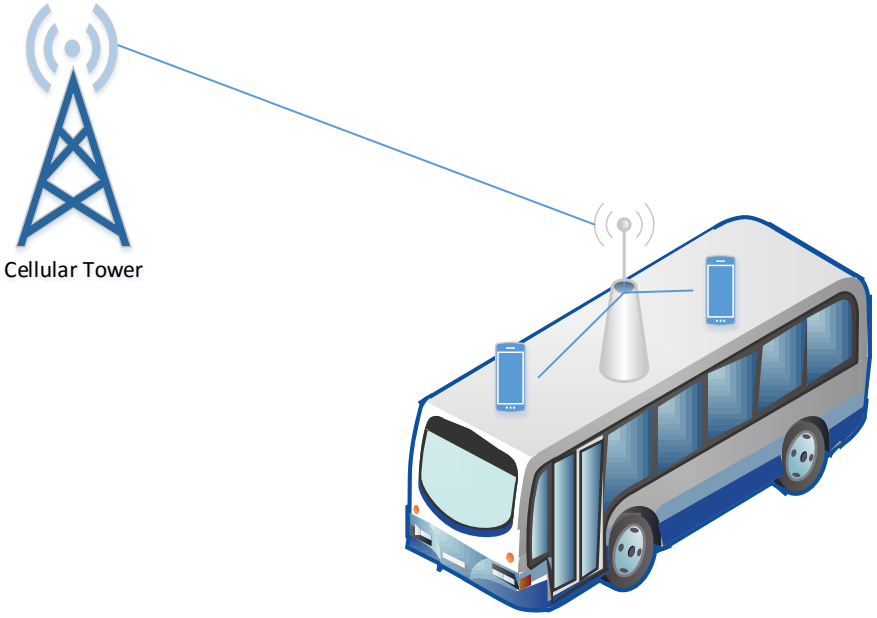


Figure 1.4 – WiFi bus solution

possible data mules (even in an ox cart) in DakNet. On the contrary, in urban public transportation networks, a public bus passes through a series of bus stops. It is way too much to install APs at all bus stops. Thus, we need to carefully select a subset of bus stops to be equipped with APs so that it can exchange a large amount of data.

The UMassDieselNet testbed [30] ran on 35 buses in Amherst, Massachusetts. It aimed at providing a real DTN environment for evaluating DTN related algorithms and protocols. This work has led to the design of the well known MaxProp routing protocol [30]. Each communication in this network is vehicle-to-vehicle communication. Each time two vehicles meet, they exchange some assistance values (that indicates the order in which packets are transmitted and deleted) and packets with one another. The motivation of our work is different as we envision the PTN to become a content provider for its customers. And instead of leveraging vehicle to vehicle communications, we focus on the design of a fixed infrastructure to push and receive content to bus customers. In other words, no direct inter-vehicle communications are performed here and buses only communicate with the PTN access network while waiting at selected bus stops.

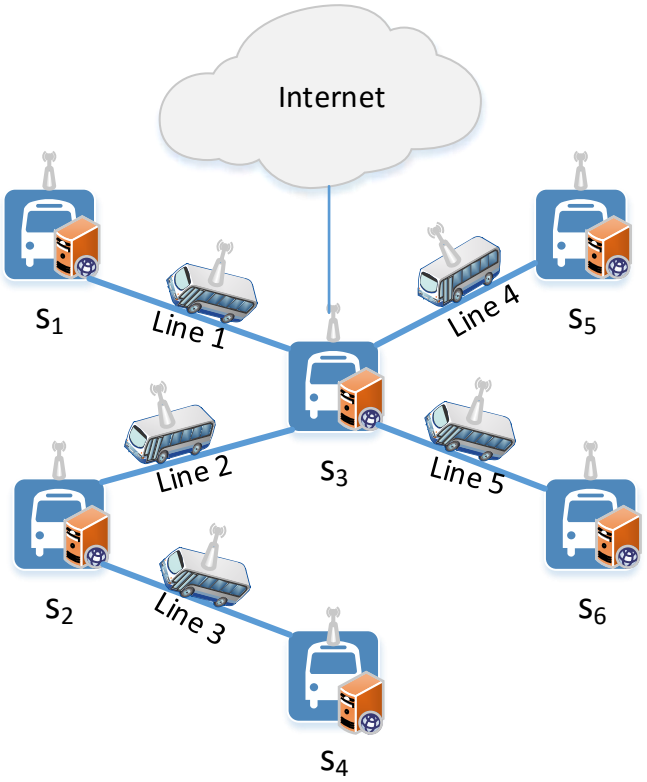


Figure 1.5 – A wireless infrastructure leveraging PTNs

More specifically, we propose a novel wireless infrastructure that leverages existing public transportation networks (PTNs). As depicted in Figure 1.5, our solution proposes to install WiFi access points on both public bus stations (some of them have Internet access) and buses. Latter ones are used as data mules, creating a delay tolerant network (DTN) capable of carrying content users can access while using the public transportation. A detailed description is given in Chapter 2.

As our infrastructure constitutes a delay tolerant network, we focus below on major works related to DTNs with a special emphasis on DTNs where node's trajectories are predictable.

1.3 Delay Tolerant Networks

Benefiting from the delay-tolerant nature of non-realtime applications, a remarkable amount of such mobile contents can be offloaded away from cellular

networks through delay tolerant networks [15, 18]. Examples of such networks have been introduced in Section 1.2.2.

1.3.1 Background on DTNs

Delay-tolerant networks are partitioned wireless ad-hoc networks with intermittent connectivity. DTNs are never fully connected at any point in time, but points of disconnection may be predictable as in vehicular networks following transportation schedules or networks with satellites traversing orbit. Communications in such networks often rely on the mobility of nodes to route messages and bridge partitions [31].

DTN provides interoperable communications with and among extreme and performance-challenged environments where continuous end-to-end connectivity cannot be assumed [32, 33, 34]. Examples of such environments include spacecraft, military/tactical, some forms of disaster response, underwater, and some forms of ad-hoc sensor/actuator networks [34]. We present below some of the most important characteristics of DTNs [32, 35, 36, 37, 38].

Sparse. The participating nodes in DTN have a very low average number of neighbors and suffer from frequent partitions. The network may never have an end-to-end path and transfer of data is done with the help of the mobility of nodes.

Dynamic. A DTN is a time-varying network where nodes move. In some settings, nodes can move independently in any direction (e.g., random walk) or in an organized fashion (e.g., according to the roadside, traffic light, highway and so on in vehicular ad-hoc networks). Communication between nodes can occur only when they are within transmission range of each other. The contact duration between mobile nodes usually lasts for a very short period, limiting time to exchange messages among nodes. Also, data transfer while nodes move results in high packet loss.

Intermittent connection. DTNs lack of continuous network connectivity which means that end-to-end paths between a pair of nodes are generally not available. Messages in DTNs can be opportunistically routed towards the destination through intermittent connections. Thus, DTN applications have to be tolerant of large delay.

Resource-constrained devices. Devices in DTNs may be resource-constrained, such as limited bandwidth, buffer, battery and computing capability.

Applications of DTNs are those operating in mobile or extreme terrestrial environments, or planned networks in space. Besides VANETs introduced in Section 1.2.1, we present below some of other most popular applications of DTNs [24, 39, 40, 41].

Interplanetary internet. The interplanetary Internet (IPN) is consisting of a set of network nodes that can communicate with each other in space. Communication would be greatly delayed by regular disconnections and the great interplanetary distances. DTN was originally designed to cope with such communications over long distances and through time delays.

Military applications. DTN is favored by the military due to its adaptability to highly changing and challenging environments, such as battlefields. Nodes (e.g., tanks, airplanes, soldiers) form a military ad-hoc network, which can be used for the transmission of delay tolerant data (e.g., data collection, reporting, etc.).

Rural communication. As Information and Communication Technology (ICT) plays an important role in developing countries, many projects are built on DTN aimed at bringing ICT to rural areas that lack communication infrastructure. Such projects include ICT4B (ICT for Billion) [42] sponsored by NSF (National Science Foundation) , TIER (Technology and Infrastructure for Emerging Regions) and E-mail4B (an e-mail system for the developing world) [43].

1.3.2 DTNs created by PTNs

As introduced in Section 1.2.2, our content delivery infrastructure is composed of fixed infrastructures (e.g., bus stops) and data mules (e.g., public vehicles), both of which are equipped with wireless transceivers and using mobile nodes as data mules, creating a delay tolerant communication network. All nodes in the network can be categorized into two types: *i*) fixed nodes, such as bus stations or stops; *ii*) mobile nodes, such as buses, tramway or subway cars. Except the characteristics mentioned above, DTNs created by PTNs have its special features:

- The network topology is stable. Bus stations or stops remain unchanged for a long time and the mobility of public vehicles follows a predefined schedule.

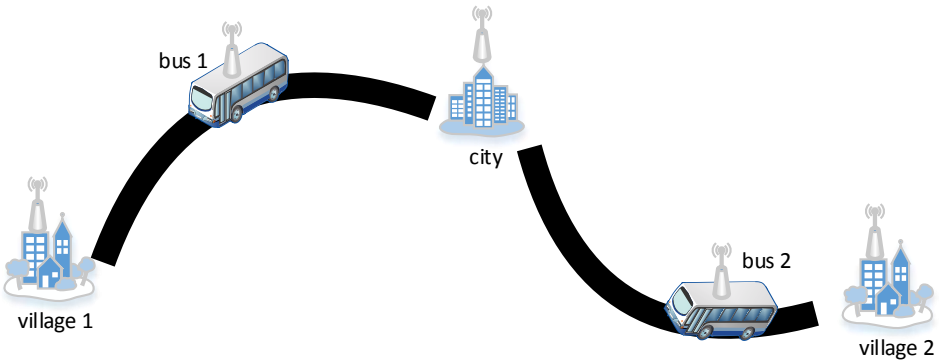


Figure 1.6 – An example of village communication networks

- The mobility of nodes are predictable. Data mules travel on a fixed schedule back and forth between two stations.

Such general DTN-based infrastructure can offer a communication service that *i*) either brings a common infrastructure to remote non-connected villages or *ii*) relieve the stress on cellular networks in dense urban areas as proposed in this PhD.

Village Communication Networks (VCN) introduced in [9] offer connectivity to remote locations in underdeveloped countries or provides a disaster-relief communication infrastructure. In this case, there is no decent communication infrastructure between remote villages and cities of a country. Communication among the inhabitants of these villages is made possible thanks to so-called data mules, as illustrated in Figure 1.6. These mules are for instance vehicles (e.g., buses, cars, etc.) carrying wireless transceivers and a given storage capacity. Each time a data mule visits a village or a city, it uploads the data dedicated to its inhabitants on a central server and downloads the data to be transferred to other remote destinations from this server. The mule carries the data from one village to another one, possibly fetching new data to a city connected to the Internet. In such a scenario, messages take time to be transferred which depends on the mules' trajectory. This communication paradigm is reminiscent of legacy mail and thus, only asynchronous communications of non-delay-sensitive data are possible. It is possible in such a scenario to leverage the network created by bus lines to provide a communication infrastructure at low cost for developing countries. Real implementations of VCNs have already been tested in the last decade. DakNet [9], developed by MIT Media

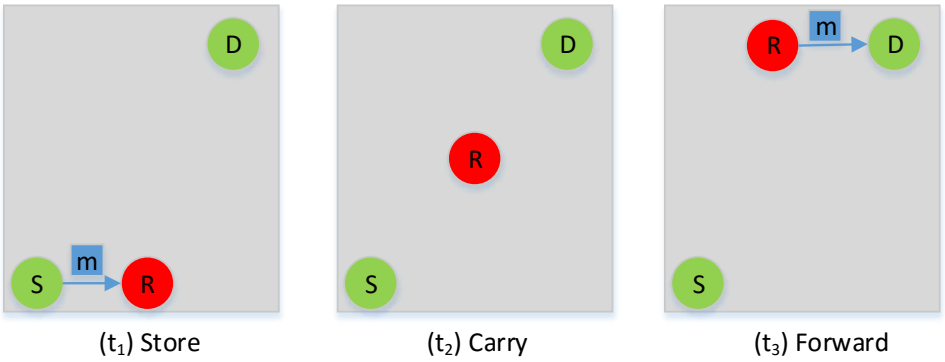


Figure 1.7 – Illustration of routing data using intermittent contacts

Lab researchers, provides low-cost digital communication for remote villages in India and Cambodia by using public buses as data mules.

In this dissertation, we envision to deploy a similar infrastructure this time in urban areas to create a novel content delivery infrastructure.

1.3.3 Routing in DTNs

Routing in DTN is a challenging problem because of the lack of continuous connectivity between distant nodes. Popular ad-hoc routing protocols such as AODV (Ad-hoc On-Demand Distance Vector) [44] or DSR (Dynamic Source Routing) [45] fail to route messages since there is no end-to-end path available at all times. Thus, specific protocols have been proposed to handle delay tolerant communications.

The simplest approach to delivering messages in DTNs is the so-called direct delivery. Messages are always held by the source node until it encounters the destination. Such an approach provides minimal overhead, but in most cases, it has the worst delivery probability. A message fails to be delivered if the source and the destination never meet within the time-to-live of the message. This can be illustrated with Figure 1.7 where the source S and the destination D never encounter each other eventually.

However, with the assistance of intermediate nodes or relay nodes, it is possible to deliver messages from a source to a destination even if there is no direct contact between them during the whole time. We continue with the example in Figure 1.7 where the relay node R makes the communication between S and D possible. R

receives a message from the source S at the time t_1, then carries it while moving to D, and finally forwards to the destination D at t_3. Therefore, routing data in delay tolerant networks usually leverages intermittent contacts between mobile entities using the store-carry-and-forward paradigm.

In such a case, transfer of data is done on a per-encounter basis: two nodes carry data and exchange it upon contact such as to rapidly diffuse a copy of it to the destination. This baseline protocol is the Epidemic protocol [46]. The protocol is similar to a viral infection process where nodes continuously replicate messages to newly discovered contacts until one copy reaches its destination. It is known to be way too resource consuming due to its elevated replication rate. Epidemic is natively better suitable for data dissemination than for routing. It provides the best delivery probability under such conditions: *i*) infinite memory at relay nodes; *ii*) sufficient contact duration between nodes. The two conditions make sure that messages between nodes can be completely exchanged. However, these strict demands make Epidemic impractical when it comes to the real world. On one hand, a limited buffer is quickly filled up with unnecessary message copies. On the other hand, the contact duration between mobile nodes may last for a very short period. There is thus not enough time to exchange all available messages among nodes. To limit the number of message replications and transfers, Spray and Wait [47] was proposed to make a trade-off between the performance and the overhead. It consists of two phrases: *i*) spray a given number of copies into the network; *ii*) wait till one of these nodes encounter the destination.

Replication-based protocols are efficient if little information on the connectivity of nodes is available. An abundance of dedicated protocols has been proposed for different scenarios in terms of network topology, mobility patterns and data flows [41, 48, 49]. The basic idea of these protocols is that the sending node carefully selects neighbor(s) as relay node(s) and message(s) to send, instead of blindly forwarding all messages to all neighbors.

Nodes in DTN may not know the availability of future encounters, but the network may benefit from learning such patterns over time [31]. A class of protocols exploits the history of previous encounters among mobile nodes to select forwarding nodes. One representative protocol is PRoPHET [50] where a value so-called delivery predictability is dynamically calculated for every pair of nodes.

Based on this information, each node only replicates messages to the neighbors with a higher delivery predictability that indicates higher probability to reach the intended destination. MaxProp [30] is based on the same principle as ProPHET. It uses several mechanisms to define the order in which packets are transmitted and deleted.

These main routing protocols are designed when little a-priori knowledge of mobility patterns exists. However, if the mobility of nodes is known in advance, appropriate routing decisions can be made.

In our application scenario, the mobility of buses is predictable and thus, leading to even more efficient routing solutions [30] where vehicle to vehicle communications are exploited to create a data mule delay tolerant network. The difficulty with vehicle to vehicle communications, even in our predictable setting, is the harsh communication environment induced by buses moving and communicating at the same time. To favor stable and predictable transmissions, we opt for a design where buses carry data between carefully planned access points. Routing paths can be calculated a-priori knowing the bus line topology, final transfer being adjusted to actual arrival and departure dates of buses on the fly.

As all messages are relayed at bus stations, they become network-wide congestion points. To address such a challenge, we propose to leverage network coding at bus stations. A basic introduction to network coding for DTNs is given next.

1.4 Network coding for DTN

WiFi, the brand name for the IEEE 802.11 WLAN (Wireless Local Area Network) technology, uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) medium access. Bandwidth is thus divided equally among contending nodes. Because of this, our content delivery infrastructure is prone to congestion bottlenecks as public transportation networks are built around the concept of hubs where many bus lines converge. This can be illustrated by Figure 1.5. Four buses affiliated with the bus lines 1, 2, 4 and 5 cross at the station s_3. Clearly, there are 5 contending nodes (4 buses and s_3) sharing the same wireless medium. WiFi AP medium access shares equally the bandwidth among these nodes. As such, each of them gets one fifth of the communication bandwidth. To complete

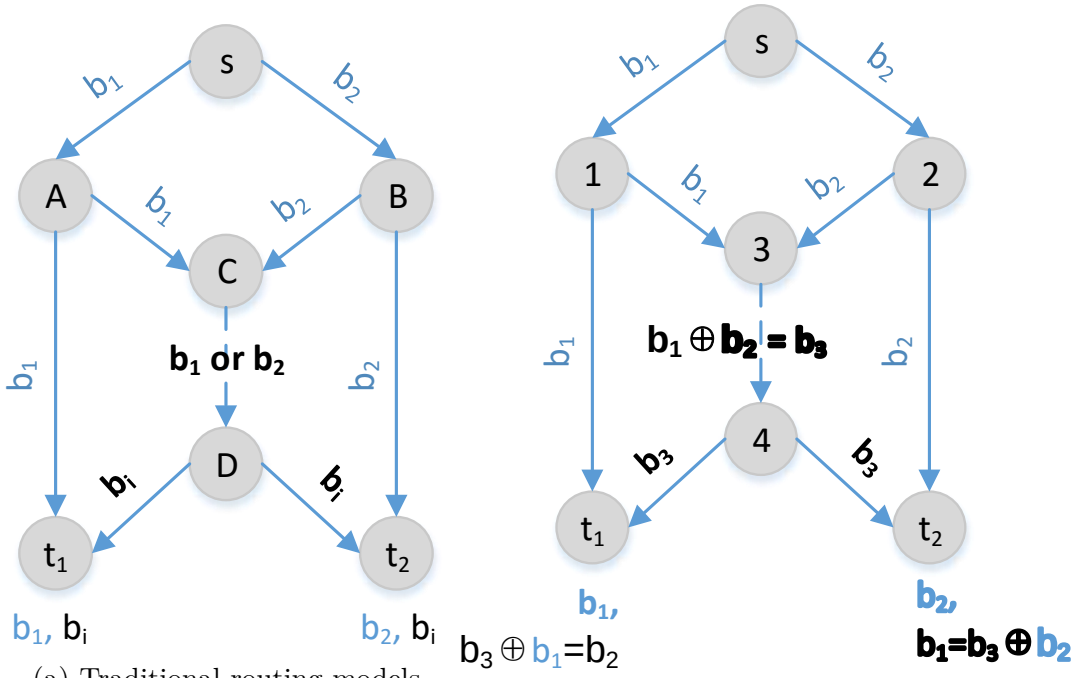
the data exchange among buses, the relay node s_3 needs as much bandwidth as the whole set of 4 buses together. In other words, s_3 is only able to relay 1/5 of packets of the 4 buses on the condition of heavy network load. Such an imbalance gets worse as the number of buses increases.

In this work, we propose to leverage network coding at bus stations to address such an imbalance.

1.4.1 Background on network coding

Network coding enables coding at the network layer: the nodes of a network combine several packets they receive together for transmission rather than simply relaying the packets [51, 52]. It has been shown to significantly improve the throughput in regular networks [53], especially in wireless networks owing to the broadcast nature of wireless medium [54]. In this thesis, we leverage network coding at the contention points of our infrastructure to relieve the imbalance of bandwidth requirements between the buses and the station.

Ahlswede, et al. proposed network coding in 2000 [51]. They characterized the admissible coding rate region for multicast sessions and their results can be regarded as the Max-flow Min-cut Theorem for network information flow. With network coding, the upper bound (the maximum achievable throughput) can be achieved in multicast scenarios. This point can be easily illustrated using the famous butterfly network depicted in Figure 1.8. The source node s aims to multicast two data bits b_1 and b_2 to both sink nodes t_1 and t_2. For the sake of discussion, we assume that the capacity of each link is a unit value, i.e. 1. It is easy to check that the value of a max-flow from s to t_l $(l = 1, 2)$ is 2. In traditional routing models, as depicted Figure 1.8-(a), the central link $C \rightarrow D$ would be only able to carry b_1 or b_2, but not both. Suppose the intermediate node C sends b_1, then t_1 receives b_1 twice but not b_2 at all. Sending b_2 poses a similar problem for t_2. The achievable rates in this case are 1.5. However, with network coding, b_1 and b_2 can be transmitted to both destinations simultaneously. As shown in Figure 1.8-(b), the intermediate node C transmits the network-coded bit b_3 that is combined from b_1 and b_2 with modulo 2 addition, i.e. $b_3 = b_1 \oplus b_2$. Thus, at t_1, b_2 can be recovered from b_1 and b_3 by calculating $b_2 = b_1 \oplus b_3$ (i.e. decode). Similarly, b_1 can be



(a) Traditional routing models

(b) With network coding

Figure 1.8 – Network coding in butterfly network

recovered at t_2. Thus the admissible rate with network coding is 2, which is equal to the max-flow capacity of 2. In summary, unlike traditional routing protocols where intermediate nodes just simply forward the exact same packets they receive, in network coding, the nodes of a network combine a set of packets (i.e. encode) into a new packet for transmission.

Following this work, Li et al. showed that linear network coding is sufficient to achieve the maximum capacity bounds in multicast traffic [52]. Linear coding regards a block of data as a vector over a certain base field and allows a node to apply a linear transformation to a vector before transmitting. Based on this work, Koetter et al. presented an algebraic framework for investigating the issues of network capacity for arbitrary networks and proposed polynomial time algorithms for encoding and decoding [55]. These results were extended to random linear network coding by Ho et al in [56, 57]. They presented a distributed random linear network coding approach for transmission and compression of information in networks.

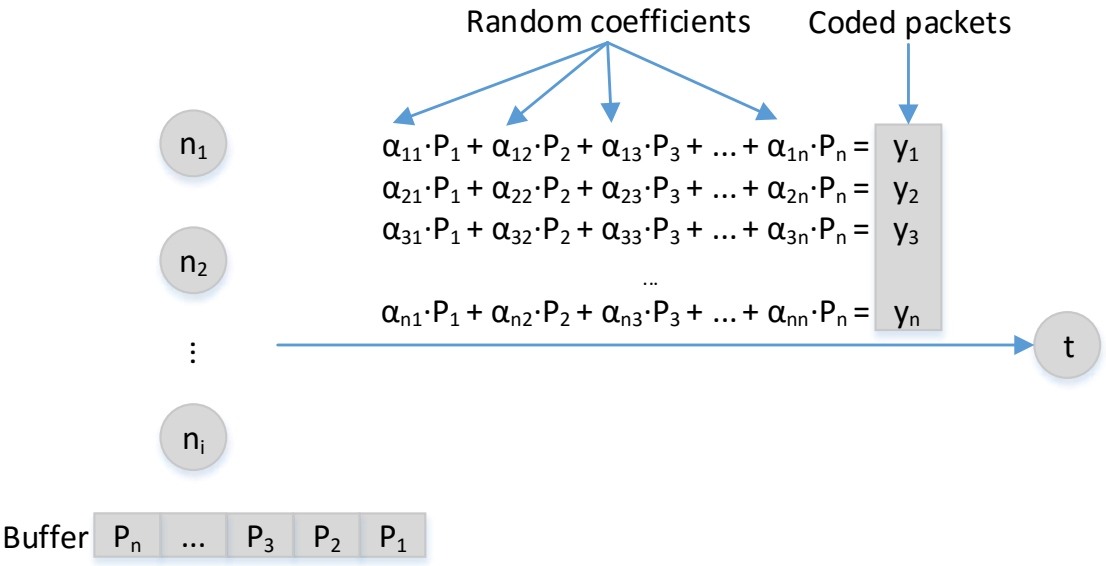


Figure 1.9 – Random linear network coding

Figure 1.9 illustrates how random linear network coding works. On the encoding side, a node select a set of packets (say $P_1, P_2, ..., P_n$) and linearly combines them together into a new network-coded packet y, $y = \sum_{i=1}^{n} \alpha_i P_i$ where network code coefficients α_i are chosen uniformly at random from a finite field $\mathbb{F}_q$ (q denotes the size of finite field). Here, information is regarded as vectors of bits which are of equal length u, represented as elements in the finite field $\mathbb{F}_q$, $q = 2^u$ [58]. It is interesting to point out that random linear network coding is degenerated into XOR network coding that is illustrated in Figure 1.8 when $q = 2$, i.e. $\mathbb{F}_2 = \{0, 1\}$. On the decoding side, the destination node t receives *enough* coded packets y_i and recovers the native packets by solving the system of linear equations Eq. (1.1) (using algorithms such as Gaussian Elimination) created by these coded packets and their coefficients.

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{1.1}$$

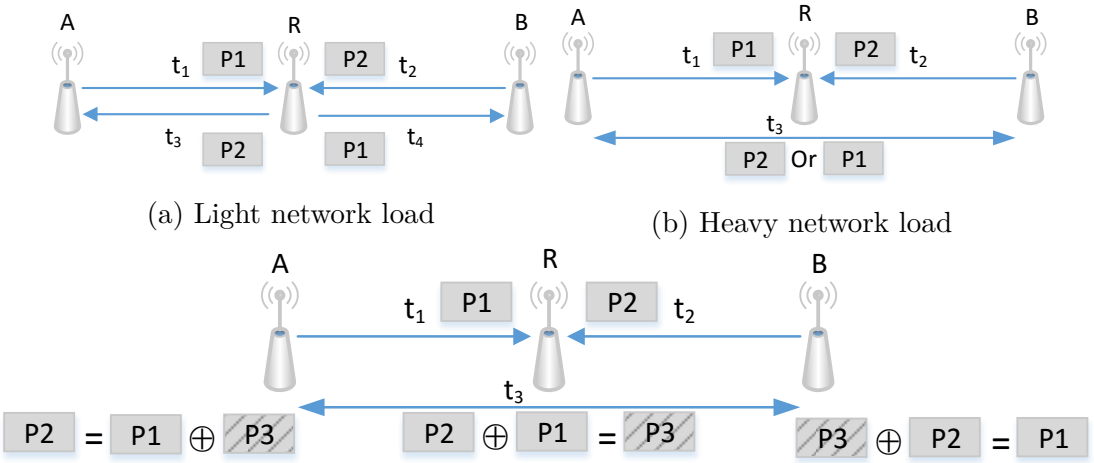
Eq. (1.1) can be simply expressed as $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ denotes the coefficient

matrix. If the matrix $\mathbf{A}$ has full rank (all n rows are independent), the system has a unique solution given by $\mathbf{x} = \mathbf{A^{-1}b}$, which means that all original packets $(P_1, P_2, ..., P_n)$ can be extracted at the node t. Formally, for a feasible multicast connection problem on a network, the probability that all the receivers can decode the source processes is at least $(1 - d/q)^\eta$ for $q > d$, where d is the number of receivers and η is the number of links carrying random combinations of source processes and/or incoming signals. This achieves capacity with probability approaching 1 with the code length u [57].

In parallel, network coding has received a lot of attention in wireless networks especially due to the shared nature of the wireless media [59]. The benefits of network coding in wireless networks can be seen from the two-way relay network in Figure 1.10 where node A and node B want to exchange a couple of packets. The radio range does not permit them to communicate directly and thus they need to go through a router R. In traditional routing methods, it takes 4 transmissions to exchange two packets $P1$ and $P2$ between them as depicted in Figure 1.10-(a). However, when it comes to heavy network load, as shown in Figure 1.10-(b), R can only drain half of packets received from A and B since R gets only a third of the channel capacity for the fair medium access. With network coding, the relay node R broadcasts the xor-ed packet $P3 = P1 \oplus P2$ in a single transmission. Both A and B can extract the desired packet by xor-ing $P3$ with the packet they have previously sent. For instance, A obtains $P2$ by calculating $P2 = P1 \oplus P3$. Similarly, $P1$ can be recovered at B. Clearly, in this case, network coding mitigates the imbalance of bandwidth requirements. Such an important improvement can be achieved for N nodes with the same relay node if each node can overhear the packets emitted by the $(N - 1)$ other nodes.

1.4.2 Network coding and DTNs

As we have explained in Section 1.3.3, main routing protocols in DTNs are replication-based as messages are allowed to be replicated for acquiring higher probability of message delivery. This brings new opportunities for network coding in DTN. Instead of receiving the same copies of a message several times, a node can obtain a set of coded messages with different coefficients from multiple



(a) Light network load

(b) Heavy network load

(c) With network coding

Figure 1.10 – Exchange two packets in two-way relay networks

opportunistic paths if appropriate random linear network coding scheme is applied.

The benefits of network coding in DTNs have been investigated for different scenarios [60][61][62] and will be introduced next.

1.4.2.1 Intra- and Inter-session network coding

One session (i.e. flow) is defined here by the sequence of messages sent by a source node to one (unicast) or more destination nodes (multicast). If only messages that belong to the same session are network coded together by the relay nodes, we refer to *intra-session* network coding. In contrast, *inter-session* network coding mixes messages that belong to different sessions together.

MORE and COPE are representative intra- and inter-session network coding approaches for wireless networks respectively, developed by MIT. In MORE [63], source nodes encode messages going to the same destination using random linear network coding. The destination recovers the original messages from a set of coded messages by solving the linear system created by these coded messages and their coefficients (i.e. decode). In COPE [54], the authors aim to improve the throughput of wireless mesh networks by using XOR network coding. Unlike traditional approaches in which intermediate nodes simply act as store and forward routers, relay nodes in COPE combine packets from different sessions using the XOR oper-

ator and then forward the XOR-ed packet to the next-hop in a single transmission. To increase the network coding probability, COPE relies on opportunistic listening where nodes snoop on all communications over the wireless medium and store the overheard packets for a limited period of time.

Recent works have presented intra- and inter-session network coding solutions for delay tolerant networks [61, 64, 65, 66, 67], where mobility of nodes is non-predictable. Most of these works have focused on people-centric DTNs where several unicast communications between mobile pairs of nodes exist. Gains in throughput have been shown most of the time for nodes moving with homogeneous mobility models and when there are high resource constraints (e.g. limited bandwidth and buffer). But this is often at the cost of an increased computation complexity or an important signaling load. Recent solutions have leveraged social routing information to improve the coding procedure at relays [67]. The authors show how challenging it is to apply inter-session random network coding to people-centric DTNs without information on the underlying network topology. In the proposed solution, authors exploit properties of the social graph (given by the SimBet protocol) to improve the routing performance but at the cost of having all nodes memorizing and updating social graph structure information.

In this thesis, we aim to gain in throughput for all the flows of the bus network and focus on multiple unicast sessions or data flows that are the dominant traffic today. Our work proposes a simple implementation of XOR network coding reminiscent of the solution proposed by Kreishah et al. in [64]. We perform pairwise inter-session XOR network coding. Messages crossing at bus stations are simply xor-ed together $m_c^{xor} = \oplus_{i=1}^{K} m_i$, reducing the bandwidth share imbalance. Details on our implementation are given in Chapter 3.

1.4.2.2 End-to-end and hop-by-hop network coding

Network coding can be implemented in two different ways: *i*) messages are encoded at the source or relay nodes and only decoded at the destination, which we refer to as *end-to-end* network coding; *ii*) messages are encoded at each hop and decoded at the next hop, which we refer to as *hop-by-hop* network coding.

For end-to-end network coding, complex decisions have to be made to select the

messages to encode together such as to maximize the probability of decoding at the destination. End-to-end network coding approach was mostly discussed for DTNs, especially in multicast or broadcast scenarios [60, 68]. For unicast applications in DTNs, the paper [61] investigates the benefits brought from random linear network coding (RLNC). Their scheme achieves better group delivery delay for a single group of packets originating from a same source and going to a same destination but at the cost of a large number of network transmissions.

On our side, we perform hop-by-hop network coding where messages are decoded immediately at the next hop rather than the destination node so as to increase the coding probability, reduce the message delay and the complexity of replication control. Hop-by-hop network coding has been successfully implemented in practice for wireless mesh networks in COPE [54]. However, the opportunistic listening of COPE can't be leveraged in our infrastructure. Buses can't hear all ongoing communications as they are not present at all times at the station. But it is easy to apply the XOR scheme of COPE to pairwise communications crossing [64] at a node of the network. We adapt this XOR scheme to our content delivery network by exploiting the features of PTNs.

1.5 Conclusion

This chapter has introduced the background of this thesis. Designing routing protocols for DTNs is challenging. After investigating existing routing protocols, we learn that the basic idea of these protocols is how the sending node selects neighbor(s) as relay node(s) and message(s) to send. For our content delivery infrastructure, we propose to employ the features of PTNs to design our routing policy. To address the congestion points and gain in throughput for multiple unicast data flows in PTNs, we propose to leverage inter-session XOR hop-to-hop network coding.

2 A Content Delivery Infrastructure Leveraging Urban PTNs

In this chapter, we present the details of our content delivery infrastructure. To address the challenge arising from the bus stop selection, we analyze the publicly available traces of PTN providers to identify the bus stops that have long contact duration. The results show that public buses stop at intermediate bus stops for a very short period. In contrast, more than 50% of the buses stop at end-of-the-line stops for more than 9 minutes. Thus, we only deploy wireless APs at the final bus stops.

In terms of routing policy, we exploit the underlying network topology of PTNs and propose that messages be delivered following the shortest path by pre-calculating routing tables for each router. It can lead to a significant reduction of message replications and transfers. We corroborate this point through simulations using real traces from three major European cities: Paris, Helsinki and Toulouse. The results show that, compared with the Epidemic routing, our scheme improves by 4 to 8 times the number of delivered messages while reducing the overhead ratio.

2.1 Scenario descriptions

Based on the observation that a significant part of the mobile content is consumed in urban areas while people are commuting using public transportation, we propose a novel content delivery infrastructure to confront with the rapid urbanization together with the looming bandwidth crunch in urban centers.

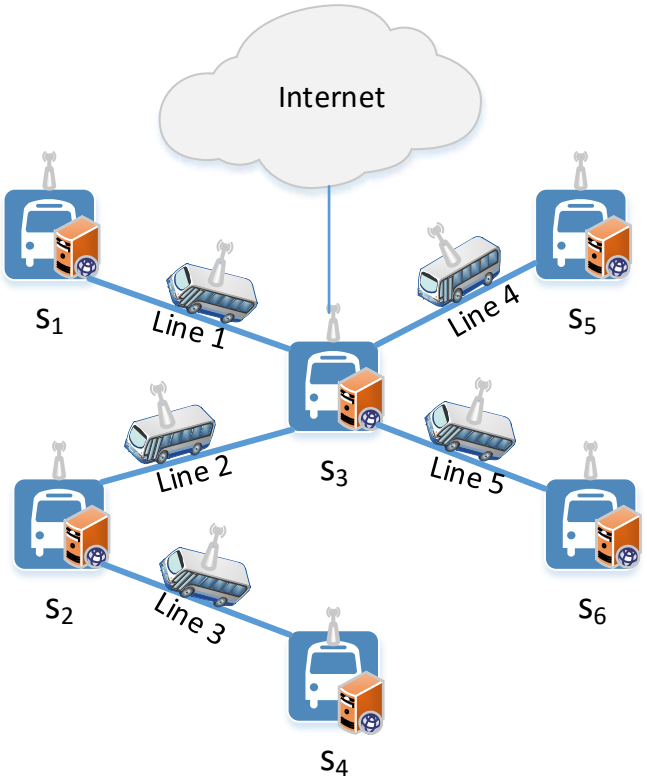


Figure 2.1 – Content delivery using PTNs.

The content delivery infrastructure envisioned in this work leverages public transportation networks. The solution, depicted in Figure 2.1, proposes to install wireless access points (e.g., WiFi) on buses and bus stops. Some of the bus stops have Internet access. Public vehicles such as buses act as data mules and carry content that PTN customers can access over wireless on the bus or waiting at bus stops. Thus, each bus is equipped with a wireless access point (AP) and local data storage capabilities. Data is updated onboard buses when they connect to wireless access points deployed at relevant bus stops of the network. Similarly, data stored on vehicles can be pushed to the bus stop AP to be routed or disseminated further in the network. Of course, AP enabled bus stops are equipped as well with storage capabilities.

Mobile users connect to the platform on the bus to download different kinds of content, including videos, books, news, etc., and can as well publish new content. In such a scenario, messages take time to be transferred which depends on the

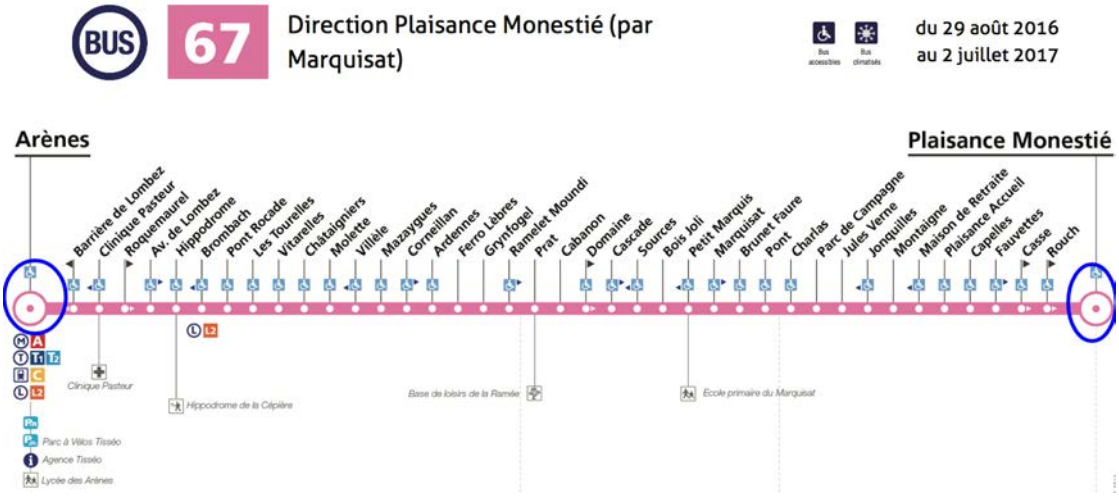


Figure 2.2 – Bus line 67 in Toulouse, France.

mules' trajectory. This communication paradigm is reminiscent of legacy mail and thus, only asynchronous communications of non-delay-sensitive data are possible.

It is interesting to note that this type of content delivery infrastructure can be leveraged as well to offload delay-tolerant data from regular access networks (e.g., cellular, ADSL) to remote and poorly connected locations using long range bus transportation networks, as introduced in Section 1.3.2.

2.2 Where to install wireless APs

Bus drivers drive their vehicles to pick up passengers and drop them off on a predetermined schedule. Typically, the vehicles pass through a series of bus stops. As illustrated in Figure 2.2, there are more than 30 intermediate stops on bus line 67 in Toulouse.

It is way too much to deploy wireless APs at all bus stops. We basically want to install APs on the bus stops that can exchange a large amount of data with buses to reduce cost. The longer a bus stop is in contact with buses, the more data the stop can relay. We analyze the publicly available traces of PTN providers to identify the bus stops that have long contact duration with buses. Three different PTNs serve as case studies in this work: a small-scale PTN from the city of

Toulouse, and two large-scale ones for Paris and Helsinki, respectively [1]. All of the traces are in GTFS (General Transit Feed Specification), a common format for public transportation schedules. Before diving into the analysis of the traces, we introduce the GTFS format for the city of Toulouse in France.

2.2.1 GTFS

The General Transit Feed Specification (GTFS), developed by Google, defines a common format for public transportation schedules and associated geographic information. It is widely used on Google Maps and other Google applications that show transit information [69].

A GTFS feed is a collection of at least 6 (required), and up to 13 CSV files [2] contained within a .zip file. The dependency among all possible files in GTFS is plotted in Figure 2.3 [3], representing the structure of GTFS. For the sake of brevity, Table 2.1 only lists the descriptions of the files [4] [69] referred to this work.

routes.txt. A city public transportation network is composed of a lot of routes or bus lines that are defined in *routes.txt*, each of which is keyed by the route ID *route_id*. For instance, there are 106 bus lines in Toulouse PTN. As an illustration, we retrieve the content of route 67 in Toulouse from *routes.txt* as shown in Listing 2.1 [5].

trips.txt. A route (time-independent) is made up of one or more trips. Trips (defined in *trips.txt*) are time-specific, each of which represents a journey taken by a vehicle and occurs at a specific time. Continuing the example of the bus line 67 (*route_id* = 11821949021891662) in Toulouse, its two trips are listed in Listing 2.2 where *direction_id* indicates the direction of travel for a trip (0 for

1. The traces are available to the public.
Toulouse: https://data.toulouse-metropole.fr/explore/dataset/tisseo-gtfs/
Paris: http://dataratp.opendatasoft.com/explore/dataset/offre-transport-de-la-ratp-format-gtfs/
Helsinki: http://developer.reittiopas.fi/pages/en/other-apis.php
2. Each record of a comma-separated values (CSV) file is consisting of one or more fields separated by commas.
3. Image from `https://en.wikipedia.org/wiki/General_Transit_Feed_Specification`.
4. The file *calendar_dates.txt* is optional in GTFS.
5. Wrap long lines with the delimiter ↪, rightwards arrow with hook. *route_type* = 3 represents Bus.

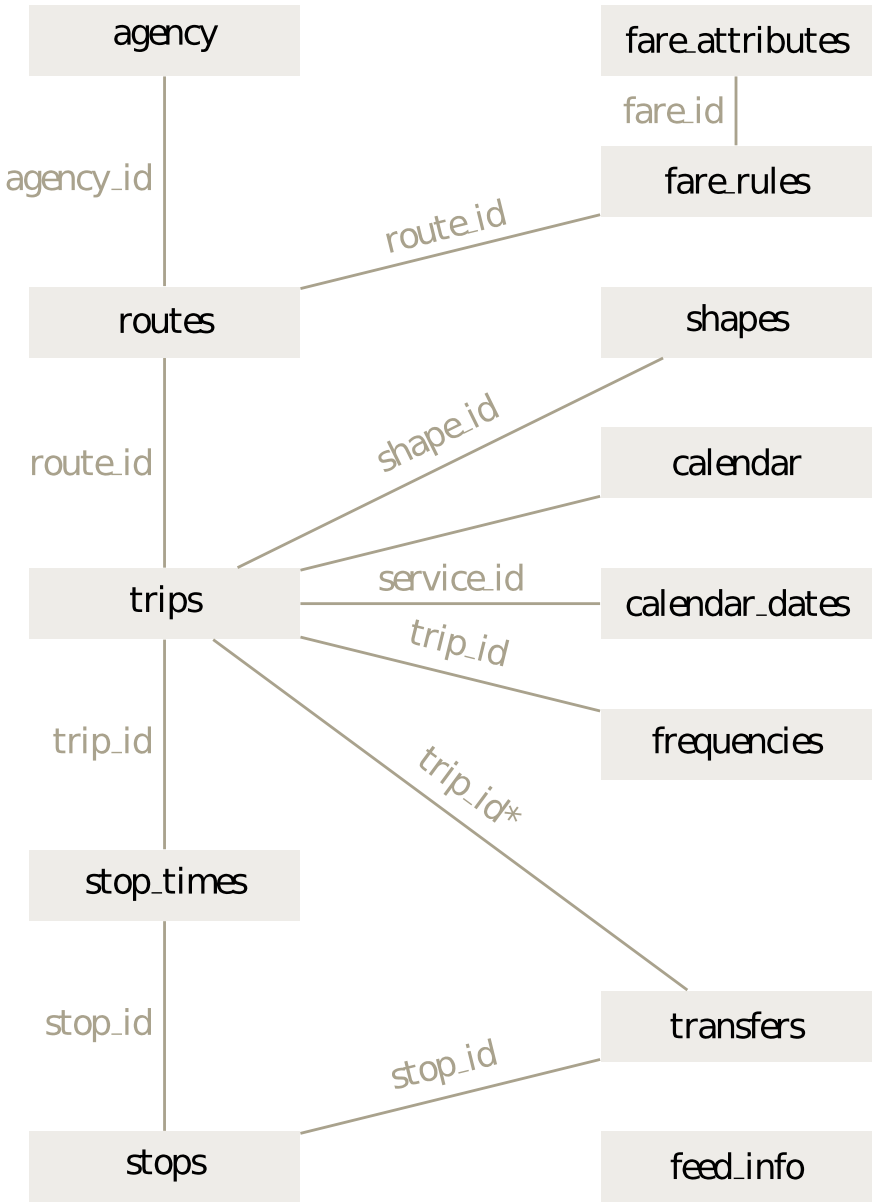


Figure 2.3 – Class diagram of GTFS.

Filename	Descriptions
routes.txt	Transit routes. A route is a group of trips that are displayed to riders as a single service.
trips.txt	Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.
stop_times.txt	Times that a vehicle arrives at and departs from individual stops for each trip.
stops.txt	Individual locations where vehicles pick up or drop off passengers.
calendar.txt	Specify the date range when service starts and ends, as well as days of the week where service is available.
calendar_dates.txt	Explicitly activate or disable for the service IDs (defined in *calendar.txt*) by date.

Table 2.1 – The descriptions of files in GTFS.

Listing 2.1 – An illustration of *routes.txt*: bus route 67 in Toulouse.

```
route_id,agency_id,route_short_name,route_long_name,route_desc,
    ↪ route_type,route_url,route_color,route_text_color
11821949021891662,6192449487677451,67,Arènes / Plaisance Monestié (par
    ↪ Marquisat),Ligne Arènes / Plaisance Monestié (par Marquisat),3,,
    ↪ ffa8ff,FFFFFF
```

Listing 2.2 – An illustration of *trips.txt*: two trips of route 67 in Toulouse

```
trip_id,service_id,route_id,trip_headsign,direction_id,shape_id
4503603928281106,4503603928278521,11821949021891662,Arènes TOULOUSE
    ↪ ,0,4503603928268191
4503603928281076,4503603928278521,11821949021891662,Plaisance Monestié
    ↪ PLAISANCE-DU-TOUCH,1,4503603928268151
...
```

Listing 2.3 – An illustration of *stop_times.txt*: the schedules of the trip 4503603928281106 in Toulouse.

```
trip_id,stop_id,stop_sequence,arrival_time,departure_time,stop_headsign,
    ↪ pickup_type,drop_off_type,shape_dist_traveled
4503603928281106,3377699720881533,0,06:20:00,06:20:00,,0,1,
4503603928281106,3377699720884041,1,06:21:00,06:21:00,,0,0,
4503603928281106,3377699720884038,2,06:22:00,06:22:00,,0,0,
4503603928281106,3377699720881531,3,06:22:00,06:22:00,,0,0,
4503603928281106,3377699720884125,4,06:23:00,06:23:00,,0,0,
4503603928281106,3377699720884123,5,06:24:00,06:24:00,,0,0,
...
4503603928281106,3377699720881134,33,06:47:00,06:47:00,,0,0,
4503603928281106,3377704015495587,34,06:48:00,06:48:00,,0,0,
4503603928281106,3377699720881127,35,06:49:00,06:49:00,,0,0,
4503603928281106,3377699723347641,36,06:50:00,06:50:00,,0,0,
4503603928281106,3377704015495711,37,06:50:00,06:50:00,,1,0,
```

outbound travel and 1 for inbound travel).

stop_times.txt. Typically, bus drivers drive their vehicles passing through a series of bus stops, as illustrated in Figure 2.2. They pick up passengers and drop them off on a predetermined schedule which is defined in *stop_times.txt*. Listing 2.3 illustrates the schedules of the trip 4503603928281106 in Toulouse. The trip passes through 38 bus stops, departure from the first bus stop at 06 : 20 and arrival at the last stop at 06 : 50.

stops.txt. A bus stop is a location where vehicles stop to pick up or drop off passengers. All bus stops are defined in the file *stops.txt*, each of which is keyed by *stop_id*. Listing 2.4 lists the corresponding stops in Listing 2.3. Multiple stops can be grouped together into a so-called parent station denoted by *parent_station*

stop_id	stop_code	location_type	stop_name	parent_station
1970324837184714		1	Arènes	
3377699720880799	180	0	Arènes	1970324837184714
3377699720880801	185	0	Arènes	1970324837184714
3377699720880802	184	0	Arènes	1970324837184714
3377699720880803	183	0	Arènes	1970324837184714
3377699720880805	188	0	Arènes	1970324837184714
3377699720880806	189	0	Arènes	1970324837184714
3377699720880808	40071	0	Arènes	1970324837184714
3377699720880809	40070	0	Arènes	1970324837184714
3377699722770341	60001	0	Arènes	1970324837184714
3377699723347641	181|2693	0	Arènes	1970324837184714
3377699723478058	60002	0	Arènes	1970324837184714
3377704015495711	182	0	Arènes	1970324837184714
3377704015496317	179	0	Arènes	1970324837184714

Figure 2.4 – An illustration of multiple stops grouped into a parent station.

($location\_type = 1$). Public transport hubs may include bus stops, metro stations, tram stops or train stations. This can be illustrated from Figure 2.4 retrieved from the GTFS trace, showing that the parent station Arènes contains 13 stops.

calendar.txt. The calendar table defines the availability of trips on weekdays within a range of dates. By way of illustration, Listing 2.5 lists the details of service ID 4503603928278521. The trips (e.g., the first trip in Listing 2.2) associated with this service ID are available on weekdays (from Monday to Friday) from February 23 to June 23, 2015. A single service can be applied to multiple different trips. If a given vehicle has different schedules, such as one schedule on weekdays and a different one on weekends, two trips (two *trip_id*) should be defined with different services (two *service_id*). Taking special dates (e.g., holiday) into consideration, bus services are usually run to different timetables. For instance, there are four service types in Toulouse: Weekday, Saturday, Sunday and Holidays, Weekday School Holidays.

calendar_dates.txt. The file *calendar.txt* defines a range of days available for trips. However, there are maybe exceptions in particular public holidays (e.g., International Workers' Day or Labour Day) during the range of days. For instance, as shown in Listing 2.5 (the second service), May 1, 2015 (Friday) falls within the date range of $service\_id = 4503599630773892$. To supplement this, the file

Listing 2.4 – An illustration of *stops.txt*: stops of the trip 4503603928281106 in Toulouse.

```
stop_id,stop_code,stop_name,stop_lat,stop_lon,location_type,
    ↪ parent_station,wheelchair_boarding
3377699720881533,15731,Plaisance Monestié
    ↪ ,43.5631,1.2909,0,1970324837185061,1
3377699720884041,15911,Fauvettes,43.5615,1.29273,0,1970324837186582,2
3377699720884038,15901,Capelles,43.5634,1.2958,0,1970324837186581,1
3377699720881531,15701,Plaisance Accueil
    ↪ ,43.5647,1.29738,0,1970324837185060,1
3377699720884125,15891,Maison de Retraite
    ↪ ,43.5637,1.29933,0,1970324837186629,1
3377699720884123,15881,Montaigne,43.5626,1.30226,0,1970324837186628,2
...
3377699720881134,291,Av. de Lombez,43.5945,1.41317,0,1970324837184870,2
3377704015495587,391,Clinique Pasteur
    ↪ ,43.5953,1.4175,0,1970329131942110,1
3377699720881127,491,Barrière de Lombez
    ↪ ,43.5952,1.41939,0,1970324837184867,1
3377699723347641,181|2693,Arènes,43.5932,1.41856,0,1970324837184714,2
3377704015495711,182,Arènes,43.5938,1.41861,0,1970324837184714,1
```

Listing 2.5 – An illustration of *calendar.txt*.

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,
    ↪ start_date,end_date
4503603928278521,1,1,1,1,1,0,0,20150223,20150623
4503599630773892,1,1,1,1,1,0,0,20140901,20151231
```

Listing 2.6 – An illustration of *calendar_dates.txt*.

```
service_id,date,exception_type
4503599630773892,20150501,2
```

calendar_dates.txt defines specific days when a trip is available or not available. In other words, the *calendar_dates* table explicitly activates or disables for the service IDs (defined in *calendar.txt*) by date. As illustrated in Listing 2.6, the service of $service\_id = 4503599630773892$ is defined as unavailable on Labor Day in 2015. The field *exception_type* indicates whether a service is available on a given date, i.e. 1 and 2 indicates that the service has been added and removed respectively for the specified date.

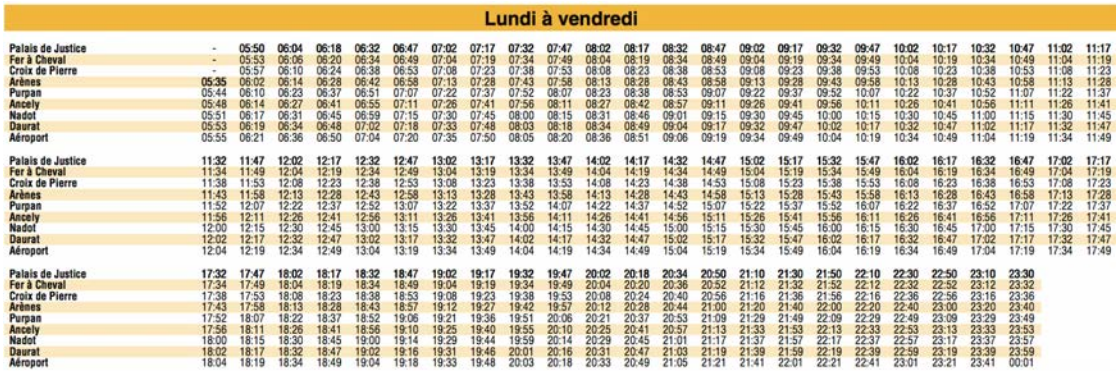
A bus route is composed of multiple trips, which usually need more than one transit bus. However, as seen above, bus ID is not recorded in GTFS trace. In order to calculate the contact duration between bus stops and buses, we first assign bus ID for each trip according to schedules. This is also necessary to generate mobility model for our later simulation in Section 2.5.1.

2.2.2 Assign bus ID for trips

In a public transportation network, a few bus routes have multiple departure locations and/or destinations. By way of illustration, Figure 2.5 shows a part of the timetable of bus route T2 in Toulouse. Clearly, T2 has two departure locations, *Arènes* and *Palais de Justice*. Actually, most of the trips have the same departure and arrival location, for instance, *Palais de Justice* and *Aéroport* in T2. For the sake of brevity, we discard those few exceptional trips. Thus, in our case, each bus route has only a departure location and a destination. In other words, a bus route is associated with two bus stops, the first and the last bus stop which in the rest of this thesis we refer to as *end stations* or simply stations to distinguish them from bus stops.

A bus ID is assigned to each trip using the schedules available in the traces. For instance, if there is a record in the dataset of a bus trip from the station s_i to s_j for the bus route r, a new bus id is assigned to this trip if there is no bus available at the station s_i for r. Otherwise, the earliest arrival bus of r at s_i is assigned to this trip.

Figure 2.6 illustrates how to assign bus ID for bus route 34 in Toulouse. We retrieve all departure times or arrival times at the end stations of the trips associated with bus route 34 from *stop_times.txt* and categorize them into 4 parts



Lundi à vendredi

Palais de Justice	-	05:50	06:04	06:18	06:32	06:47	07:02	07:17	07:32	07:47	08:02	08:17	08:32	08:47	09:02	09:17	09:32	09:47	10:02	10:17	10:32	10:47	11:02	11:17
Fer à Cheval	-	05:53	06:06	06:20	06:34	06:49	07:04	07:19	07:34	07:49	08:04	08:19	08:34	08:49	09:04	09:19	09:34	09:49	10:04	10:19	10:34	10:49	11:04	11:19
Croix de Pierre	-	05:57	06:10	06:24	06:38	06:53	07:08	07:23	07:38	07:53	08:08	08:23	08:38	08:53	09:08	09:23	09:38	09:53	10:08	10:23	10:38	10:53	11:08	11:23
Arènes	05:35	06:02	06:14	06:28	06:42	06:58	07:13	07:28	07:43	07:58	08:13	08:28	08:43	08:58	09:13	09:28	09:43	09:58	10:13	10:28	10:43	10:58	11:13	11:28
Purpan	05:44	06:10	06:23	06:37	06:51	07:07	07:22	07:37	07:52	08:07	08:23	08:38	08:53	09:07	09:22	09:37	09:52	10:07	10:22	10:37	10:52	11:07	11:22	11:37
Ancely	05:48	06:14	06:27	06:41	06:55	07:11	07:26	07:41	07:56	08:11	08:27	08:42	08:57	09:11	09:26	09:41	09:56	10:11	10:26	10:41	10:56	11:11	11:26	11:41
Nadot	05:51	06:17	06:31	06:45	06:59	07:15	07:30	07:45	08:00	08:15	08:31	08:46	09:01	09:15	09:30	09:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45
Daurat	05:53	06:19	06:34	06:48	07:02	07:18	07:33	07:48	08:03	08:18	08:34	08:49	09:04	09:17	09:32	09:47	10:02	10:17	10:32	10:47	11:02	11:17	11:32	11:47
Aéroport	05:55	06:21	06:36	06:50	07:04	07:20	07:35	07:50	08:05	08:20	08:36	08:51	09:06	09:19	09:34	09:49	10:04	10:19	10:34	10:49	11:04	11:19	11:34	11:49
Palais de Justice	11:32	11:47	12:02	12:17	12:32	12:47	13:02	13:17	13:32	13:47	14:02	14:17	14:32	14:47	15:02	15:17	15:32	15:47	16:02	16:17	16:32	16:47	17:02	17:17
Fer à Cheval	11:34	11:49	12:04	12:19	12:34	12:49	13:04	13:19	13:34	13:49	14:04	14:19	14:34	14:49	15:04	15:19	15:34	15:49	16:04	16:19	16:34	16:49	17:04	17:19
Croix de Pierre	11:38	11:53	12:08	12:23	12:38	12:53	13:08	13:23	13:38	13:53	14:08	14:23	14:38	14:53	15:08	15:23	15:38	15:53	16:08	16:23	16:38	16:53	17:08	17:23
Arènes	11:43	11:58	12:13	12:28	12:43	12:58	13:13	13:28	13:43	13:58	14:13	14:28	14:43	14:58	15:13	15:28	15:43	15:58	16:13	16:28	16:43	16:58	17:13	17:28
Purpan	11:52	12:07	12:22	12:37	12:52	13:07	13:22	13:37	13:52	14:07	14:22	14:37	14:52	15:07	15:22	15:37	15:52	16:07	16:22	16:37	16:52	17:07	17:22	17:37
Ancely	11:56	12:11	12:26	12:41	12:56	13:11	13:26	13:41	13:56	14:11	14:26	14:41	14:56	15:11	15:26	15:41	15:56	16:11	16:26	16:41	16:56	17:11	17:26	17:41
Nadot	12:00	12:15	12:30	12:45	13:00	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45
Daurat	12:02	12:17	12:32	12:47	13:02	13:17	13:32	13:47	14:02	14:17	14:32	14:47	15:02	15:17	15:32	15:47	16:02	16:17	16:32	16:47	17:02	17:17	17:32	17:47
Aéroport	12:04	12:19	12:34	12:49	13:04	13:19	13:34	13:49	14:04	14:19	14:34	14:49	15:04	15:19	15:34	15:49	16:04	16:19	16:34	16:49	17:04	17:19	17:34	17:49
Palais de Justice	17:32	17:47	18:02	18:17	18:32	18:47	19:02	19:17	19:32	19:47	20:02	20:18	20:34	20:50	21:10	21:30	21:50	22:10	22:30	22:50	23:10	23:30		
Fer à Cheval	17:34	17:49	18:04	18:19	18:34	18:49	19:04	19:19	19:34	19:49	20:04	20:20	20:36	20:52	21:12	21:32	21:52	22:12	22:32	22:52	23:12	23:32		
Croix de Pierre	17:38	17:53	18:08	18:23	18:38	18:53	19:08	19:23	19:38	19:53	20:08	20:24	20:40	20:56	21:16	21:36	21:56	22:16	22:36	22:56	23:16	23:36		
Arènes	17:43	17:58	18:13	18:28	18:43	18:57	19:12	19:27	19:42	19:57	20:12	20:28	20:44	21:00	21:20	21:40	22:00	22:20	22:40	23:00	23:20	23:40		
Purpan	17:52	18:07	18:22	18:37	18:52	19:06	19:21	19:36	19:51	20:06	20:21	20:37	20:53	21:09	21:29	21:49	22:09	22:29	22:49	23:09	23:29	23:49		
Ancely	17:56	18:11	18:26	18:41	18:56	19:10	19:25	19:40	19:55	20:10	20:25	20:41	20:57	21:13	21:33	21:53	22:13	22:33	22:53	23:13	23:33	23:53		
Nadot	18:00	18:15	18:30	18:45	19:00	19:14	19:29	19:44	19:59	20:14	20:29	20:45	21:01	21:17	21:37	21:57	22:17	22:37	22:57	23:17	23:37	23:57		
Daurat	18:02	18:17	18:32	18:47	19:02	19:16	19:31	19:46	20:01	20:16	20:31	20:47	21:03	21:19	21:39	21:59	22:19	22:39	22:59	23:19	23:39	23:59		
Aéroport	18:04	18:19	18:34	18:49	19:04	19:18	19:33	19:48	20:03	20:18	20:33	20:49	21:05	21:21	21:41	22:01	22:21	22:41	23:01	23:21	23:41	00:01		

Figure 2.5 – The timetable of bus route T2 in Toulouse.

in ascending order (by time) as shown in the four gray boxes (the first column) in Figure 2.6. We check the departure times one by one in ascending order. For instance,

- **06:00:00**. Begin with the earliest departure time. Assign a new bus id (say 0) to the trip as there is no bus available at *UPS* at that time.
- **06:01:00**. Assign a new bus id (say 1) to the trip as there is no bus available at *Arènes* at that time.
- **06:22:00**. Use the bus id 0 for the trip as bus 0 is already in *Arènes* (arrived at 06 : 18 : 00).
- **06:25:00**. Use the bus id 1 for the trip as bus 1 is already in *UPS* (arrived at 06 : 19 : 00).
- ...

With the above steps, each trip of bus line 34 is assigned with a bus id, as shown in the second column of the four gray boxes in Figure 2.6. The blue arrows indicate the journeys of bus 0. Such a method can be applied to other bus routes. Finally, each trip in the trace is associated with a bus id.

2.2.3 Trace subset selection

A subset of the traces is selected to evaluate our propositions and to derive statistics using simulations. For each city, we only select the period ranging from 7:00 to 19:00 which corresponds to the regular service period of most bus lines. We avoid the exceptional service related to special dates (e.g., holidays), we select

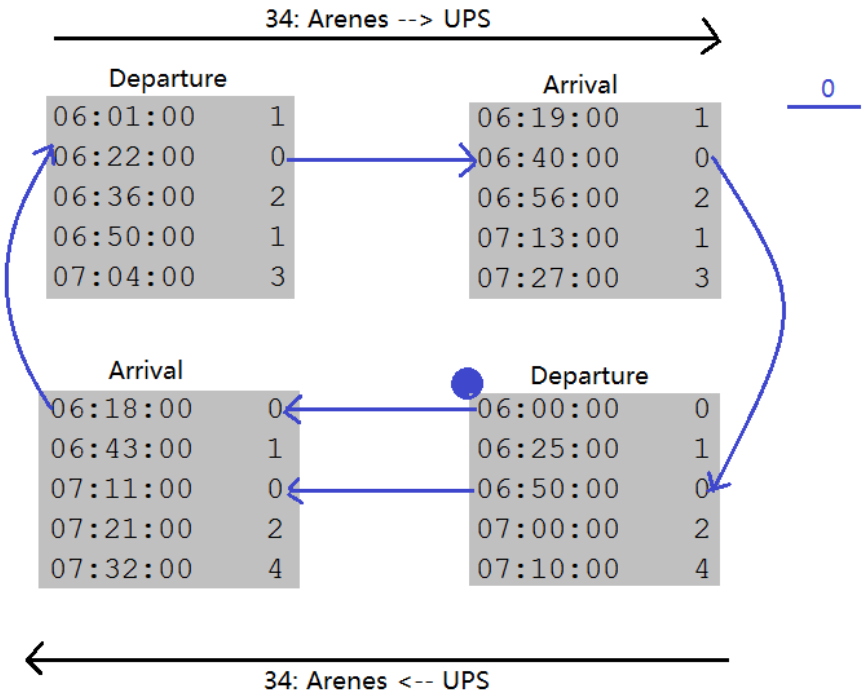


Figure 2.6 – Assign bus ID for the trips of bus route 34 in Toulouse.

the trace of a working day for each city.[1]. With this selection, we are able to get all available services for this day according to *calendar.txt* and *calendar_dates.txt*. It is easy to get all available trips for this day as each trip relates to a service id.

From this trace subset, a graph G representing available bus routes (derived by the available trips) is defined. Edges represent bus lines and vertex represent corresponding end stations. In this thesis, our derivations hold for a connected graph. As such, it G is composed of several connected components, we select the largest connected component of G.

To sum up, we use a subset of the trace satisfying the following conditions: *i*) The period ranges from 7:00 to 19:00; *ii*) The service type is working days; *iii*) The graph induced by the subset is connected.

1. Toulouse: 2015-03-18, Wednesday
Paris: 2016-03-01, Tuesday
Helsinki: 2016-05-11, Wednesday

2.2.4 Bus stop selection

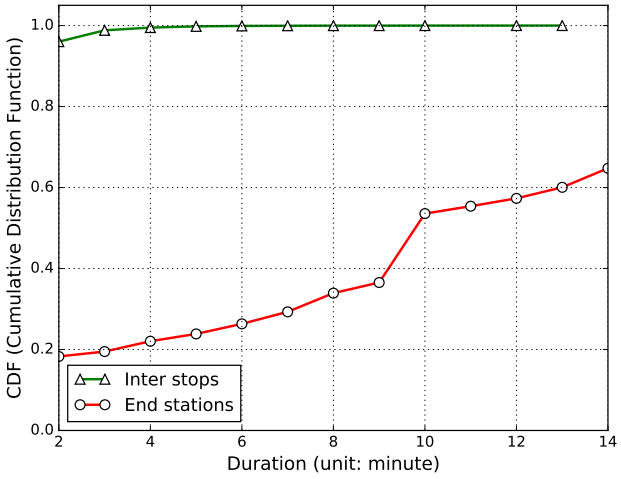
Our goal is to deploy wireless APs at bus stops where buses stop for an extended period of time to favor the exchange of large files. However, the contact duration of a bus stop with the buses passing through it is not recorded in our traces. This can be easily seen from *stop_times.txt* where the departure time and the arrival time are always the same, as illustrated in Listing 2.3.

It is possible to calculate the travel time of a bus between two consecutive bus stops to estimate the contact duration between the bus and its passing intermediate stops. For instance, if a bus travels from bus stop s_1 to s_2 at time t_1 and t_2, the travel time is $(t_2 - t_1)$. Consequently, the time the bus waits at s_1 is strictly less than $(t_2 - t_1)$. The cumulative distribution function (CDF) of the travel time of buses between two consecutive bus stops is given in Figure 2.7 under the label "Inter stops". For 90% of the cases, the CDS shows that travel times between consecutive stops are within three minutes. And thus, buses wait obviously less than three minutes at bus stops.

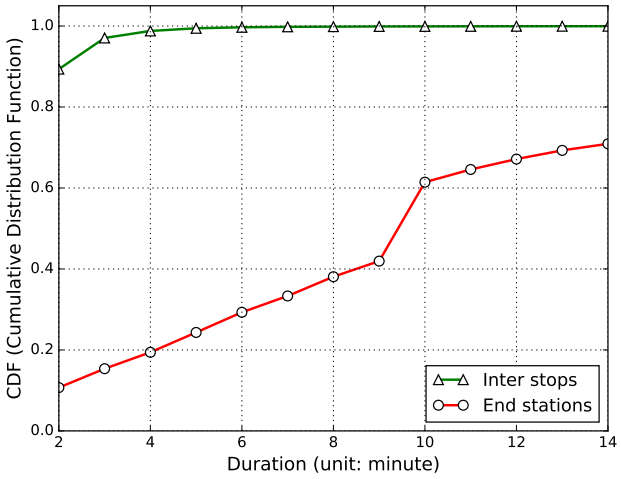
After assigning a bus ID for each trip, it is easy to calculate the contact duration of a bus with end stations. For instance, if a bus arrives at a station at time t_1 and departs at time t_2, its waiting time at the station equals $(t_2 - t_1)$. For the first trip and the last trip of each bus route, we assume their waiting time is 10 minutes. We plot the CDF of the waiting time of the bus at an end station in Figure 2.7 under the label "End stations". It shows that more than 50% of the buses stop at the end stations for more than 9 minutes.

The above analysis results are consistent with our intuition. Public vehicles acting as data mules travel on a fixed schedule back and forth between two end stations, stopping along the way at different intermediary locations for a very short period of time. In contrast, at end stations, conductors usually rest for an extended period of time before engaging in the next journey. Moreover, multiple bus lines usually cross at such stations. This results in an extended contact duration between bus lines at end stations.

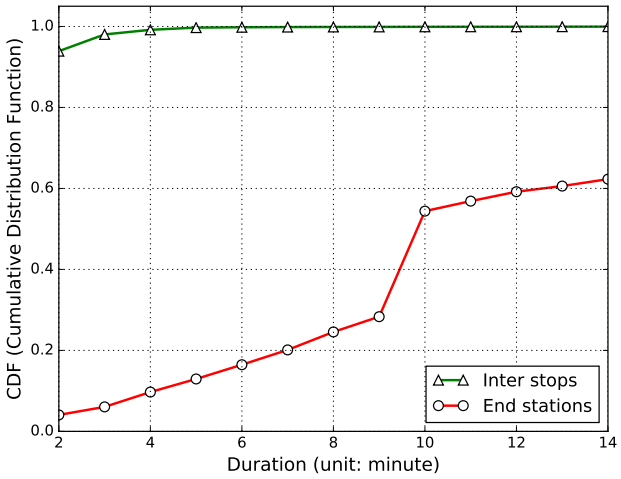
It is beneficial to leverage such extended contact duration for exchanging data between the buses. Thus, end stations are particularly interesting elements of the PTN. In our architecture, we only deploy wireless access points at the end



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 2.7 – The CDF of the duration of inter stops and waiting time at end stations.

stations of bus lines and do not consider intermediary stops for such purpose. For instance, only the end stations, *Arènes* and *Plaisance Monestié* as circled in blue in Figure 2.2, are equipped with wireless APs in T2.

To sum up, the main advantages of our proposed content delivery infrastructure are as follows:

- Low cost. The infrastructure relies on inexpensive WiFi technology and an extensive public transportation network already in place. Only end stations have to be equipped with wireless APs.
- Low packet loss. Communication between buses and the station is done statically when public vehicles park at end stations rather than move.
- Large message data transmission. The waiting time of buses at end stations last a couple of minutes or even more, which makes transferring large contents such as videos possible.

2.3 Models and assumptions

2.3.1 PTN model and topologies

As explained in Section 2.2, only the end stations of bus lines are equipped with wireless APs. Thus, we only need to consider the end stations of bus routes and can skip their intermediate stops.

Almost all networks support bidirectional communication working under half-duplex or full-duplex and such networks are represented by undirected graphs. Thus, we model our content delivery infrastructure as an undirected graph $G(V, E)$ where edges represent bus lines and vertices represent corresponding end stations. Formally, there is an edge $e = (s_i, s_j)$ in E if there exists a bus service between two end stations s_i and s_j. Each bus b_k of the PTN is associated with an edge (s_i, s_j), a pair of end stations.

As pinpointed earlier, our derivations hold for a connected graph. If G is not connected, all computations can be applied to the individual connected components of the PTN of interest. Examples of such graphs are given in Figure 2.8, showing the largest connected components of three different PTNs of Toulouse, Paris and Helsinki. Table 2.2 lists the main characteristics of these networks. For

City	Nodes	Edges	Buses
Toulouse	44	46	297
Paris	213	236	3056
Helsinki	217	266	1512

Table 2.2 – Investigated PTN topologies

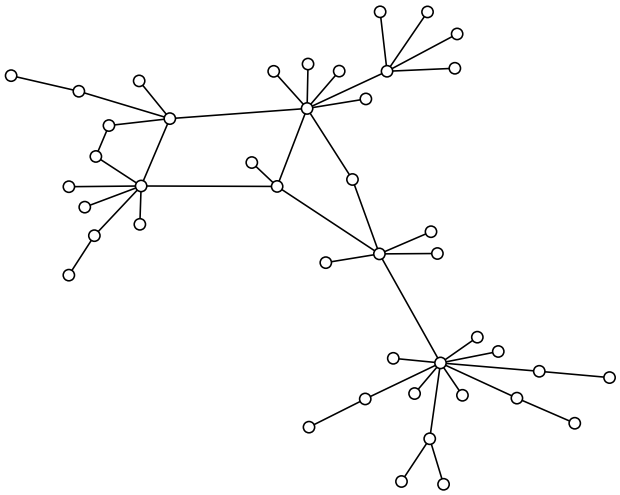
instance, the biggest component of Toulouse contains 44 end stations, 46 bus lines and 297 public vehicles running on it.

2.3.2 Traffic model

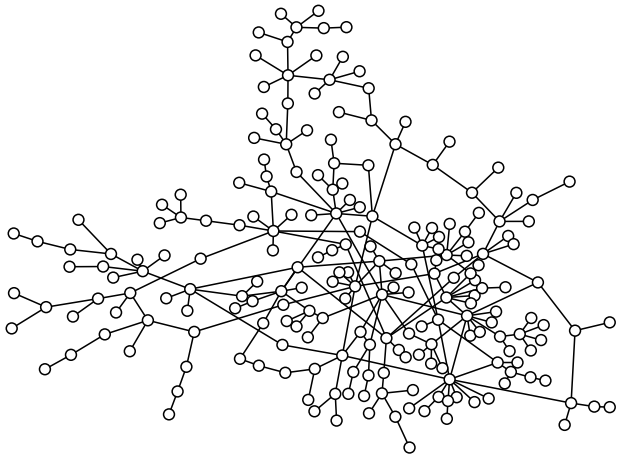
Users of this content delivery service connect to a platform on the bus that offers different pieces of content (videos, books, news, etc.). It is possible to update the content available on the bus using the delay tolerant network described earlier. Users may *i*) publish new content to the onboard platform which can be spread to other nodes of the network, or *ii*) subscribe to new content to be fetched from another node. Content not available in the PTN may be obtained from a node connected to the Internet. To save deployment costs, only one or two end stations have an Internet access.

This thesis leaves for further investigation how content is actually updated, requested and fetched. The aim of this thesis is to show the pure networking benefit of using PTNs for efficiently carrying delay tolerant content in a cost-efficient deployment.

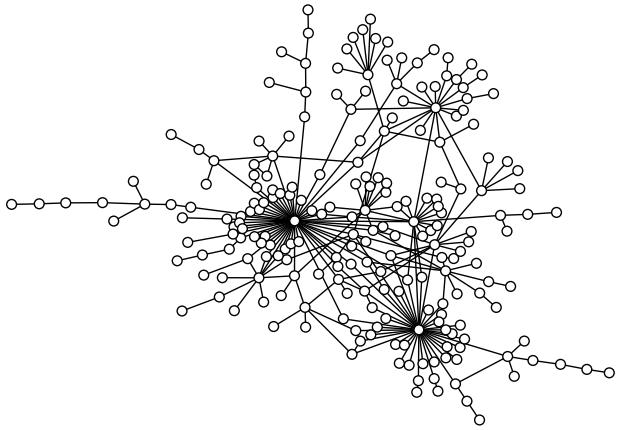
As such, we simplify the network traffic model by assuming that each node of the network pushes a constant flow of messages to be routed into the PTN. Messages are generated periodically at every end station with a creation period of Δ time units. Every time a new message is created by end station s_i, its destination end station s_j ($j \neq i$) is selected at random among possible ones. With such a traffic model, we are able to capture the maximum throughput the PTN offers to all possible flows of the network.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 2.8 – The biggest connected component of public transportation networks.

2.3.3 Medium access control

There is no doubt that the most widely used WLAN (Wireless Local Area Network) technology, commonly referred to as WiFi, follows the IEEE 802.11 standards that implement the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) medium access protocol. CSMA/CA divides the channel equally among all transmitting nodes within the collision domain.

To be fair, we assume in this work that a medium access control at the base station divides the bandwidth equally between the contending nodes. Thus, any fair medium access control mechanism such as CSMA (Carrier Sense Multiple Access) or TDMA (Time Division Multiple Access) can be implemented in practice. For TDMA, the derivations presented next hold if a single slot is given to all nodes requiring service. In other words, if B buses are willing to exchange data with the AP, one slot is given to each bus and one to the AP (for a total of $B + 1$ slots). In the rest of this manuscript, all propositions are illustrated for CSMA.

2.4 Our routing policy

Our content delivery infrastructure boils down to a delay tolerant network (DTN) as there is no end-to-end path between any two end stations at any time. Due to the lack of continuous connectivity in DTNs, specific protocols leveraging the store-carry-and-forward paradigm have been proposed to handle delay tolerant communications. Replication-based protocols become mainstream as messages are allowed to be replicated, leading to an increase of the number of messages delivered, but at the cost of an increase of network resource consumption (e.g., bandwidth, energy and storage). Sophisticated mechanisms have been proposed to limit the number of message replications and transfers.

As stated in Section 1.3, main routing protocols in DTN are designed for non-predictable mobility patterns. However, in our application scenario, the underlying topology of PTN remains unchanged for a long time and the mobility of buses are predictable. Knowing these features, better performing routing can be made.

In this work, we define routing tables at each end station to deliver messages following the predefined path. Routes can be calculated a-priori knowing the

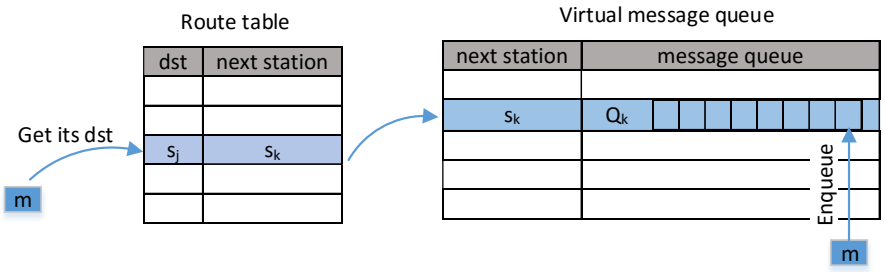


Figure 2.9 – Message reception at a station.

bus line topology as illustrated in Figure 2.8 by using Dijkstra's shortest path algorithm. Thus, a route between s_i and s_j is given by the sequence of stations $(s_i, s_{i+1}, ..., s_j)$ that minimizes the number of buses used. Corresponding routing tables are stored at the end stations. As shown in Figure 2.9, each record consists of two fields $(dst, next\_station)$ where dst represents the address of the destination and $next\_station$ represents the address of the next station to which the packet has to be sent on the way to its final destination. Note that no bus ID is required here as several buses convey passengers on the same route.

When a bus arrives at a station, it uploads as many messages as possible to the station's AP until it leaves, possibly sharing the bandwidth with other buses. After receiving a message m from a bus, as illustrated in Figure 2.9, the station extracts m's destination s_j and looks up its next hop in the routing table, say s_k. Then, m is placed into a virtual queue Q_k that stores only packets going to next hop s_k.

In parallel, the station tries to empty the messages stored in its queues to the set of buses currently connected to its access point. As depicted in Figure 2.10, from a list of buses B currently waiting at the station, the station extracts the set of next hop station S that can be reached through them, and corresponding virtual queues. In a round-robin manner, a message is dequeued from one of these queues and sent to the corresponding bus.

In the rest of this thesis, we refer to this routing policy as **Baseline**.

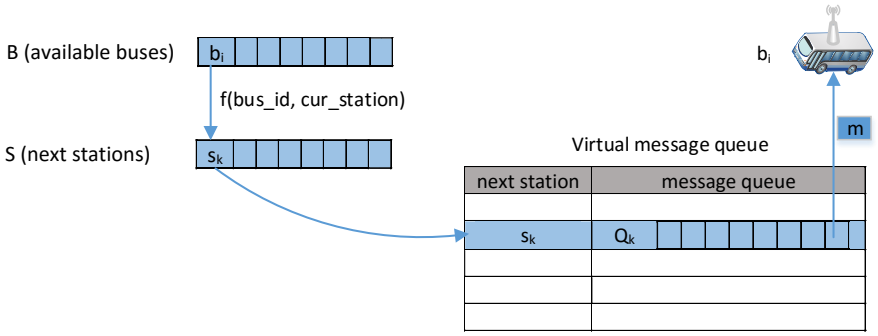


Figure 2.10 – Message emission at a station.

2.5 Performance evaluation

Performance evaluation is carried out on the ONE (Opportunistic Network Environment) simulator [70] which is specifically designed for evaluating DTN routing and application protocols. We compare our Baseline routing policy to Epidemic by measuring the number of messages delivered from 7:00 to 19:00. We measure as well the overhead ratio defined as the ratio of the number of emissions to the number of messages delivered (emissions include initial message broadcast and subsequent relay ones). Trace-based simulations show that our scheme improves 4 to 8 times the number of delivered messages and reduces the overhead ratio.

2.5.1 Simulation setup

Mobility model. Real traces of the public transportation networks of Toulouse, Paris, and Helsinki are used for this evaluation, selected so as to represent cities of different scales. We select a subset of the traces for our simulation. The descriptions and the subset selection of the traces are given in Section 2.2. It is easy to convert the subset of the traces into the ONE standard connection events. A connection is up if a bus arrives at a station, and down if a bus departs from a station.

Data flows. A message is created at every station with a given time period Δ (set to 20 seconds for the data presented here) while the simulation is running. The message destination is selected uniformly at random among all the stations

Key	Value
Mobility model	PTN traces of Toulouse, Paris and Helsinki
Simulation duration	From 7:00 to 19:00, 12 hours
Update interval	1 second
Message interval Δ	20 seconds, multiple unicast data flows
Time-to-live	12 hours
Buffer size	infinite
Message queue type	1, random

Table 2.3 – Simulation parameters.

except itself.

Routing. For all three topologies, routing tables are generated for each station before the simulation starts so that messages are relayed based on the shortest paths, as explained in Section 2.4.

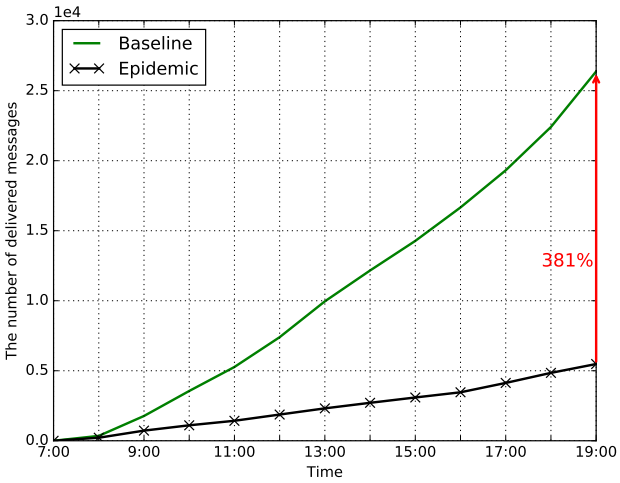
The simulation parameters are summarized in Table 2.3. In the simulations, a message is transmitted at each time unit among contending nodes. The time unit is set to one second (the update interval). In one second, only one of contending nodes accesses the wireless channel and builds up a connection with another node. During this period, a message is sent and received. To avoid the side effect of the message queue, we set: *i*) the buffer size to infinite; *ii*) the time-to-live of messages to 12 hours; *iii*) the message queue type to random (the default option in the ONE) where message order is randomized every time.

2.5.2 Evaluation of Baseline

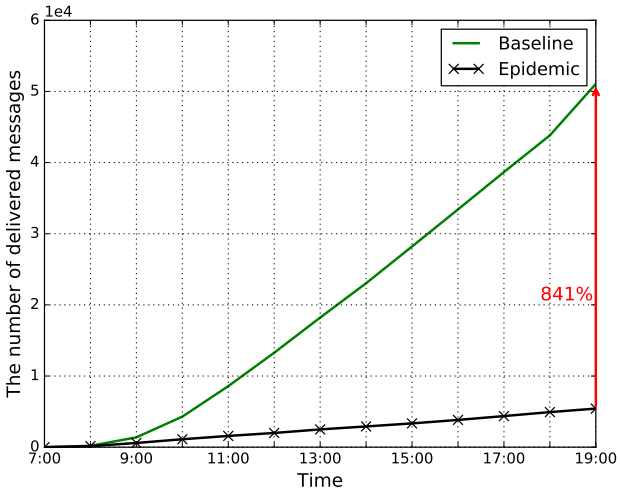
In this section, we evaluate the performance of our routing policy Baseline introduced in Section 2.4 in terms of message delivery and overhead ratio.

Method: We do statistics on the number of delivered messages per hour while the simulation is running and the number of emissions at the end of the simulation.

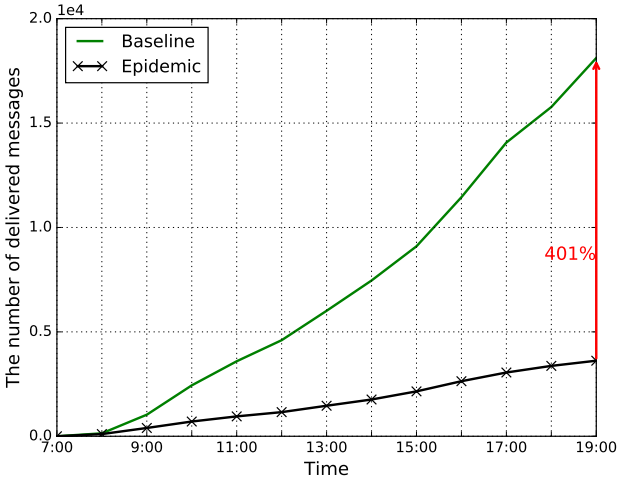
Results: Figure 2.11 shows the evolution of the number of delivered messages for Baseline and Epidemic over a period from 7:00 to 19:00. Clearly, Baseline outperforms Epidemic as expected, showing a 381%, 841% and 401% improvement in Toulouse, Paris and Helsinki respectively. These gains result from a dramatic decrease in the number of message replications and transfers.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 2.11 – Comparison of Baseline and Epidemic on the packet delivery.

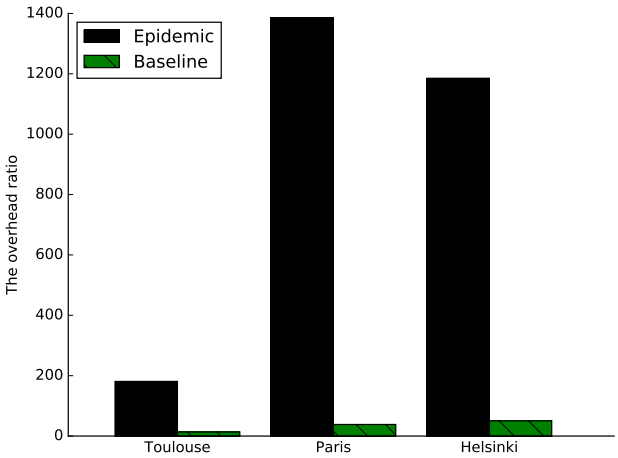


Figure 2.12 – Comparison of Baseline and Epidemic on the overhead ratio.

Epidemic is similar to a viral infection process where nodes continuously replicate and transmit messages to newly discovered contacts. It floods the whole network with unnecessary message copies. In contrast, messages in Baseline can be routed following the pre-specified path and therefore no unnecessary message copy exists in the network. For the same reason, the overhead ratio is declined significantly as demonstrated in Figure 2.12.

2.6 Conclusion

In this chapter, we present a novel content delivery infrastructure leveraging PTNs where only end stations are equipped with wireless APs to help relieve the bandwidth crunch in urban areas. It has the advantages of low cost, low packet loss and large message data transmission. Employing the stable underlying topology and predictable schedules of PTNs, we propose to pre-calculate a route table for each station so that messages can be routed following a predefined shortest path. This leads to a dramatic decrease in message replications and transfers. Simulation results on three realistic traces demonstrate that our routing policy improves 4 to 8 folds in the number of delivered messages while also reducing the overhead ratio compared to Epidemic.

3 XOR Network Coding for the Content Delivery Infrastructure

Public transportation networks are built around the concept of hubs, with many bus lines converging periodically in few major stops. Therefore, the content delivery infrastructure we introduced in Chapter 2 is prone to periodic congestion events – many buses wanting to offload their data at the same time and place. To address this challenge, we propose to leverage XOR network coding at bus stations.

Towards this, in this chapter, we introduce a simple theoretical analysis that measures the benefit of XOR network coding for our platform. We start with the simplest scenario of two bus lines converging in a station and then extend our analysis to the general case of N bus lines. Unlike similar works on XOR network coding, we need to take into account the fact that buses are not perfectly synchronized, making the network topology and the potential for network coding time sensitive. Therefore, we evaluate the impact of XOR network coding on the message delivery probability as a function of the overlapping time interval between bus lines. The analysis indicates that the message delivery probability can be increased by 50% while overhead ratio can be reduced by 50%.

Simulations using the ONE simulator [70] corroborate our theoretical results. The trace-based simulations using the public transportation networks of three European cities show that network coding improves message delivery from 35% to 48%.

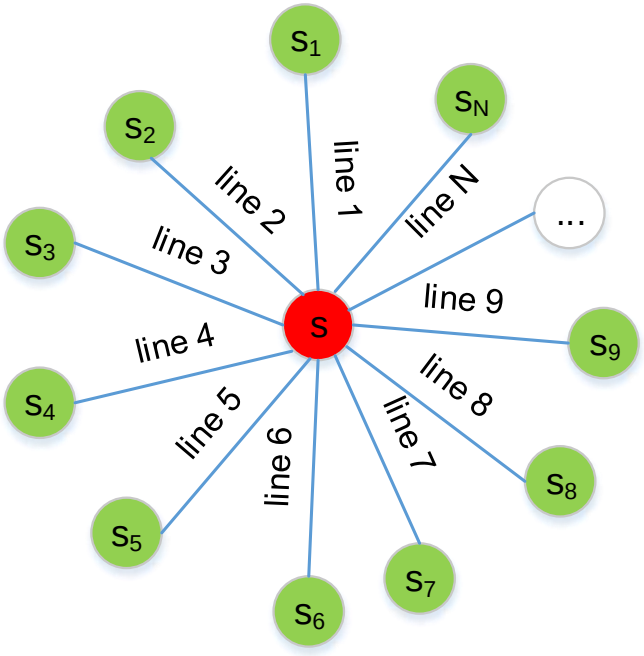


Figure 3.1 – Many bus lines converging in the same cross station.

3.1 Problem statement

We focus on the scenario depicted in Figure 3.1 where several bus lines converge in the same station, s. A public vehicle (bus) servicing a bus line is associated with two end stations, s_i and s. The stations and buses are equipped with an IEEE 802.11 wireless communication interface. When a bus connects to a *leaf* station, s_i, it uploads the data intended for s_i and downloads the data to be transferred outbound. Similarly, when it connects to the *cross* station, s, it uploads the data originating from s_i and downloads the data for its way back to s_i. A message m between two leaf nodes, s_i, s_j, is transmitted following the path $s_i \rightarrow b_i \rightarrow s \rightarrow b_j \rightarrow s_j$, where b_i and b_j denote one of the public buses associated with (s_i, s) and (s, s_j), respectively. In our analysis, we assume that the storage capacity is unbounded.

Clearly, it falls on the cross station, s, to route the messages between the different bus lines, creating a congestion point. If N_b buses converge in the cross station, $N_b + 1$ transmitters will contend simultaneously for the communication

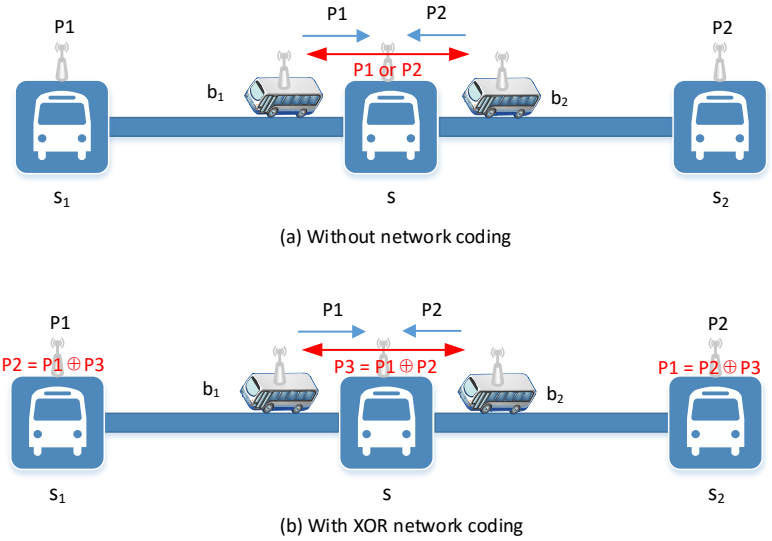


Figure 3.2 – Exchange two packets via a station.

bandwidth. Furthermore, the N_b buses all need to upload traffic to the cross station who in turn needs to push traffic to all the N_b buses. The medium access control protocol of IEEE 802.11 is designed to share the channel fairly among competing radios, regardless of their load, leading to a situation where every transmitter gets $1/(N_b + 1)$ share of the bandwidth. Therefore, the cross station only has a $1/(N_b + 1)$ bandwidth share while it has N_b times more data to send than each of the N_b buses. Obviously, under such conditions the queue at the cross station will grow over time, leading to buffer overflow and packet losses.

3.2 Using network coding for relieving congestion

To mitigate the potential for congestion points at public transportation hubs, we propose to use network coding, which has been shown to significantly improve the system throughput in such scenarios [54]. As explained in Section 1.4, we leverage *inter-session* XOR *hop-by-hop* network coding. More specifically, messages that belong to different sessions are simply xor-ed together (i.e. $m_c^{xor} = \oplus_{i=1}^{K} m_i$) and decoded at the following next hop.

Figure 3.2-(a) illustrates the standard solution to exchange packets at relay nodes. It takes four transmissions to exchange two packets, $P1, P2$ between two buses, b_1, b_2 via the station s without network coding. In this case, s needs twice as many transmissions as either b_1 or b_2. As shown in Figure 3.2-(b), if the station s broadcasts a coded packet, $P3 = P1 \oplus P2$, using a single transmission, both next hop stations, s_1 and s_2, can extract a new packet by xor-ing $P3$ with the packet they have previously sent. s_1 can obtain $P2$ by calculating $P3 \oplus P1 = P2$ and s_2 can obtain $P1$ by calculating $P3 \oplus P2 = P1$. Thus, with XOR network coding, the base station needs a single transmission instead of two, avoiding the potential for congestion and packet losses. Such an improvement can be achieved for N_b bus lines crossing at the same station as well by pairing all messages that cross at s.

3.3 Theoretical analysis

In this section, we capture analytically the benefits of network coding for relieving congestion in PTNs.

3.3.1 Scenario description

We begin with the simplest scenario where two bus lines cross at the same station, s, as depicted in Figure 3.2. In such a scenario, there exists only one pairwise flow (s_1, s_2). For the convenience of discussion, the message queue of s is separated into two virtual queues, storing the messages to be downloaded from the bus b_1 and b_2 respectively. We draft the main lines of an inter-session XOR network coding exchange protocol to analyze the performance gain compared to a basic store-and-forward mechanism.

In the second scenario, we consider the general case of N_b bus lines converging in the cross station, s, as depicted in Figure 3.1. We assume N_b is an even number and therefore the network has $N_b/2$ pairwise flows. The station s has N_b virtual message queues, creating a one-to-one correspondence between message queues, buses and leaf stations. The data uploaded by a bus is stored in the queue corresponding to its destination address.

Both scenarios are investigated assuming that all buses arrive at and leave the

Term	Description
t_l	bus waiting time at a leaf station
t_b	bus travel time
t_c	bus waiting time at the cross station
T	round-trip time of a bus, equivalent to $(2t_b + t_l + t_c)$
Δ	message creation period at leaf stations, $\Delta > 0$
L	number of messages carried by one bus, equivalent to T/Δ
N	number of delivered messages *without* network coding
N_{nc}	number of delivered messages *with* network coding
N_t	number of transmissions
N_b	number of bus lines crossing at the same station
P	delivery probability *without* network coding, defined as $N/(N_b \cdot L)$
P_{nc}	delivery probability *with* network coding, defined as $N_{nc}/(N_b \cdot L)$
G_p	delivery probability gain *with* network coding, defined as $(N_{nc}-N)/(N_b \cdot L)$
C	overhead ratio *without* network coding, defined as N_t/N
C_{nc}	overhead ratio *with* network coding, defined as N_t/N_{nc}
G_c	overhead ratio gain *with* network coding, $(C - C_{nc})/C$
t_1	time during which bus b_1 is alone at the cross station
t_{12}	time during which buses b_1 and b_2 are present at the cross station
t_2	time during which b_2 is alone at the cross station
r	ratio of overlapping interval, defined as $t_{12}/(t_1 + t_{12} + t_2)$
R	bit rate or transmission rate of base station
D	amount of data two buses exchange *without* network coding
D_{nc}	amount of data two buses exchange *with* network coding
G_t	throughput gain achieved by network coding, defined as $(D_{nc} - D)/D$

Table 3.1 – Notations and definitions

station at the same time. This assumption is later relaxed, and we examine how the delivery probability is affected by the difference in the overlapping intervals. The analysis is performed for a homogeneous setting: all buses stay for the same duration, t_l, at the leaf stations, s_i, and for the same time, t_c, at the cross station, s. They have as well the same travel time, t_b.

The notations used in this chapter are summarized in Table 3.1 with the main notations illustrated in Figure 3.3.

We analytically derive the gain in delivery probability, G_p, when performing our pairwise inter-session XOR network coding compared to the no-coding for-

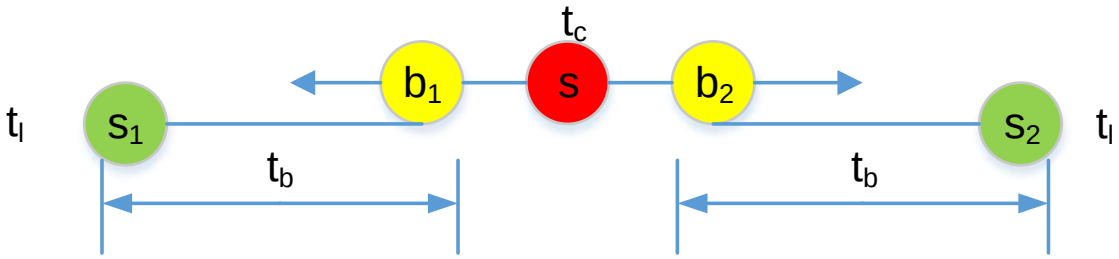


Figure 3.3 – The communication network model of two bus lines.

warding at the cross station. The formal definition of G_p is given in Table 3.1. In the following, we derive, for various scenarios, its expression as a function of the message creation period, Δ, i.e., of the network load. Here, all leaf stations create messages with the same period, Δ.

3.3.2 The two-bus-line scenario

When all nodes can make full use of the available bandwidth, the maximum number of messages each bus can transfer is $t_c/3$ with a rate of $R = 1$ message per time unit. The cross station transfers $t_c/3$ messages to two buses without network coding, but the number is doubled $(2 \cdot t_c/3)$ with network coding.

The delivery probability without network coding P, with network coding P_{nc} and the gain G_p are calculated for different message creation periods as follows:

• For $\Delta < 3 \cdot T/t_c$: In this case, more messages are created per bus line than the base station can serve with network coding, and no protocol can drain all messages. In other words, the number of messages carried by one bus, denoted by L, exceeds $t_c/3$. Thus, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = \frac{\frac{t_c}{3}}{2 \cdot \frac{2t_b+t_l+t_c}{\Delta}} = \frac{t_c}{6 \cdot T} \cdot \Delta \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = \frac{2 \cdot \frac{t_c}{3}}{2 \cdot \frac{2t_b+t_l+t_c}{\Delta}} = \frac{t_c}{3T} \cdot \Delta \\ G_p &= \frac{N_{nc} - N}{N_b \cdot L} = \frac{t_c}{6 \cdot T} \cdot \Delta \end{aligned} \tag{3.1}$$

• For $\Delta \in [3 \cdot T/t_c, 4 \cdot T/t_c)$: In this case, the cross station can only drain all

created messages if it uses network coding. Here, we have $L \leq t_c/3$ (permitting complete message drain with network coding) and $L > t_c/4$ (non complete message drain without network coding). Thus, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = \frac{t_c - 2 \cdot L}{2 \cdot L} = \frac{t_c}{2 \cdot T} \cdot \Delta - 1 \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = 1 \\ G_p &= \frac{N_{nc} - N}{N_b \cdot L} = 2 - \frac{t_c}{2 \cdot T} \cdot \Delta \end{aligned} \tag{3.2}$$

The gain reaches its maximum value, $G_p = 1/2$, for $\Delta = 3 \cdot T/t_c$.

• For $\Delta \geq 4 \cdot T/t_c$: The base station can drain all messages with or without network coding as $L \leq t_c/4$. Thus, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = 1 \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = 1 \\ G_p &= \frac{N_{nc} - N}{N_b \cdot L} = 0 \end{aligned} \tag{3.3}$$

3.3.3 The N_b-bus-line scenario

The previous analysis is extended to N_b bus lines for the pairwise inter-session XOR network coding protocol described earlier. As N_b buses connect to the cross station, $(N_b + 1)$ nodes compete for the channel with an access probability of $1/(N_b + 1)$.

In this scenario, the maximum number of messages a bus can upload is $t_c/(N_b + 1)$ with a rate of $R = 1$ message per time unit. If *no network coding* is applied, the maximum number of messages that N_b buses can download from the base station is $t_c/(N_b + 1)$. If pairwise *network coding* is applied, this number is doubled since for any two uploads, there is only one download from the cross station.

• For $\Delta \leq (N_b + 1) \cdot T/t_c$: The number of packets created L is strictly higher than $t_c/(N_b + 1)$ and thus the cross station can not drain all packets whichever

protocol is being used. In this case, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = \frac{t_c/(N_b+1)}{N_b \cdot L} = \frac{t_c \cdot \Delta}{N_b \cdot (N_b+1) \cdot T} \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = \frac{2t_c/(N_b+1)}{N_b \cdot L} = \frac{2 \cdot t_c \cdot \Delta}{N_b \cdot (N_b+1) \cdot T} \\ G_p &= \frac{N_{nc}-N}{N_b \cdot L} = \frac{t_c \cdot \Delta}{N_b \cdot (N_b+1) \cdot T} \end{aligned} \tag{3.4}$$

• For $\Delta \in \left(\frac{(N_b+1)\cdot T}{t_c}, \frac{3\cdot N_b \cdot T}{2\cdot t_c}\right)$: In this case, we have $L < \frac{t_c}{N_b+1}$ and $L > \frac{2\cdot t_c}{3\cdot N_b}$. Network coding can still not drain all messages since it can at most transmit $\frac{2\cdot t_c}{3\cdot N_b}$, but it gets more efficient than in the previous case. Here, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = \frac{t_c - L \cdot N_b}{L \cdot N_b} = \frac{t_c \cdot \Delta}{N_b \cdot T} - 1 \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = \frac{(t_c - L \cdot N_b) \cdot 2}{N_b \cdot L} = \frac{2 \cdot t_c \cdot \Delta}{N_b \cdot T} - 2 \\ G_p &= \frac{N_{nc}-N}{N_b \cdot L} = \frac{t_c \cdot \Delta}{N_b \cdot T} - 1 \end{aligned} \tag{3.5}$$

• For $\Delta \in \left[\frac{3\cdot N_b \cdot T}{2\cdot t_c}, \frac{2\cdot N_b \cdot T}{t_c}\right)$: In this case, network coding can drain all messages but regular forwarding cannot. Indeed, $L \leq \frac{2\cdot t_c}{3\cdot N_b}$ but $L > \frac{t_c}{2\cdot N_b}$, the maximum number of messages regular forwarding can manage. Here, we have:

$$\begin{aligned} P &= \frac{N}{N_b \cdot L} = \frac{t_c - L \cdot N_b}{N_b \cdot L} = \frac{t_c \cdot \Delta}{N_b \cdot T} - 1 \\ P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = 1 \\ G_p &= \frac{N_{nc}-N}{N_b \cdot L} = 2 - \frac{t_c \cdot \Delta}{T \cdot N_b} \end{aligned} \tag{3.6}$$

The gain reaches the maximum (i.e. 1/2), while $\Delta = \frac{3\cdot N_b \cdot T}{2\cdot t_c}$ where the cross node is *just* able to drain all created messages with network coding.

• For $\Delta \geq \frac{2\cdot N_b \cdot T}{t_c}$: The base station drains all created messages even without using network coding. Thus, we have,

$$
\begin{aligned}
P &= \frac{N}{N_b \cdot L} = 1 \\
P_{nc} &= \frac{N_{nc}}{N_b \cdot L} = 1 \\
G_p &= \frac{N_{nc} - N}{N_b \cdot L} = 0
\end{aligned}
\tag{3.7}
$$

3.3.4 The overhead ratio

As pointed out earlier, exchanging a pair of messages can save one transmission with network coding. We are interested in how much overhead can be saved by XOR network coding. We define the overhead ratio C as the ratio of the number of transmissions to the number of messages delivered. In the following, we analytically derive the gain in overhead ratio G_c for the two bus lines scenario when performing our pairwise inter-session XOR network coding compared to the no-coding forwarding at the cross station. The formal definition of G_c is given in Table 3.1.

The overhead ratio without network coding C, with network coding C_{nc} and the gain G_c is calculated for different message creation periods by:

• For $\Delta < 6 \cdot T/(6 \cdot t_l - t_c)$: In this case, all nodes exhaust the available bandwidth equally. This requires that enough messages are created per bus line, i.e. $L > (t_l - t_c/6)$ without network coding and $L > (t_l - t_c/3)$ with network coding. Here, the total number of transmissions is $(t_l + t_c + t_l)$. The cross station transfers $t_c/3$ messages to buses without network coding, but the number is doubled $(2 \cdot t_c/3)$ with network coding. Thus, we have:

$$
\begin{aligned}
C &= \frac{N_t}{N} = \frac{t_l + t_c + t_l}{\frac{t_c}{3}} = \frac{3 \cdot (2 \cdot t_l + t_c)}{t_c} \\
C_{nc} &= \frac{N_t}{N_{nc}} = \frac{t_l + t_c + t_l}{2 \cdot \frac{t_c}{3}} = \frac{3 \cdot (2 \cdot t_l + t_c)}{2 \cdot t_c} \\
G_c &= \frac{C - C_{nc}}{C} = \frac{1}{2}
\end{aligned}
\tag{3.8}
$$

• For $\Delta \in \left[\frac{6\cdot T}{6\cdot t_l - t_c}, \frac{3\cdot T}{3\cdot t_l - t_c}\right)$: In this case, all nodes exhaust bandwidths if network coding is applied. This requires that at least $(t_l - t_c/3)$ messages are created for each bus, i.e. $L > (t_l - t_c/3)$. But for non-network coding, leaf stations do not use up the bandwidths. Thus, we have,

$$\begin{aligned} C &= \frac{N_t}{N} = \frac{(T/\Delta + t_c/6)\cdot 2 + t_c}{\frac{t_c}{3}} = 4 + \frac{6\cdot T}{t_c \cdot \Delta} \\ C_{nc} &= \frac{N_t}{N_{nc}} = \frac{t_l + t_c + t_l}{2\cdot \frac{t_c}{3}} = \frac{3\cdot(2\cdot t_l + t_c)}{2\cdot t_c} \\ G_c &= \frac{C - C_{nc}}{C} = \frac{12\cdot T + (5\cdot t_c - 6\cdot t_l)\cdot \Delta}{12\cdot T + 8\cdot t_c \cdot \Delta} \end{aligned} \tag{3.9}$$

The overhead ratio gain reaches its maximum of $G_c = 1/2$ at $\Delta = 6\cdot T/(6\cdot t_l - t_c)$

• For $\Delta \in \left[\frac{3\cdot T}{3\cdot t_l - t_c}, \frac{3\cdot T}{t_c}\right)$: In this case, more messages are created per bus line than the base station can serve at best with network coding, and no protocol can drain all messages. In other words, the number of messages carried by one bus, denoted L, exceeds $t_c/3$. Thus, we have:

$$\begin{aligned} C &= \frac{N_t}{N} = \frac{(T/\Delta + t_c/6)\cdot 2 + t_c}{\frac{t_c}{3}} = 4 + \frac{6\cdot T}{t_c \cdot \Delta} \\ C_{nc} &= \frac{N_t}{N_{nc}} = \frac{2\cdot(T/\Delta + t_c/3) + t_c}{\frac{t_c}{3}} = \frac{5}{2} + \frac{3\cdot T}{2\cdot t_c \cdot \Delta} \\ G_c &= \frac{C - C_{nc}}{C} = \frac{6\cdot T + 3\cdot t_c \cdot \Delta}{12\cdot T + 8\cdot t_c \cdot \Delta} \end{aligned} \tag{3.10}$$

The overhead ratio gain reaches its maximum of $G_c = \frac{6\cdot t_l + t_c}{12\cdot t_l + 4\cdot t_c}$ at $\Delta = \frac{3\cdot T}{3\cdot t_l - t_c}$. For instance, $G_c = 7/16$ if $t_l = t_c$.

• For $\Delta \in [3\cdot T/t_c, 4\cdot T/t_c)$: In this case, the cross station still can not drain all created messages in the case of non-network coding. This requires that at least $4\cdot T/t_c$ messages are created for each bus, i.e. $L > t_c/4$. However, for network coding, the cross station does not use up the bandwidths. Thus, we have:

$$
\begin{aligned}
C &= \frac{N_t}{N} = \frac{T/\Delta \cdot 2 + (t_c - T/\Delta \cdot 2) + t_c}{t_c - T/\Delta \cdot 2} = \frac{2 \cdot t_c \cdot \Delta}{t_c \cdot \Delta - 2 \cdot T} \\
C_{nc} &= \frac{N_t}{N_{nc}} = \frac{T/\Delta \cdot 2 + T/\Delta \cdot 3}{T/\Delta \cdot 2} = \frac{7}{2} \\
G_c &= \frac{C - C_{nc}}{C} = \frac{14 \cdot T - 3 \cdot t_c \cdot \Delta}{4 \cdot t_c \cdot \Delta}
\end{aligned}
\tag{3.11}
$$

The overhead ratio gain reaches its maximum of $G_c = 5/12$ at $\Delta = 3 \cdot T/t_c$.

• For $\Delta \geq 4 \cdot T/t_c$: In this case, The base station can drain all messages with and without network coding as $L \leq t_c/4$. Thus, we have:

$$
\begin{aligned}
C &= \frac{N_t}{N} = \frac{T/\Delta \cdot 2 + T/\Delta \cdot 4}{T/\Delta \cdot 2} = 4 \\
C_{nc} &= \frac{N_t}{N_{nc}} = \frac{T/\Delta \cdot 2 + T/\Delta \cdot 3}{T/\Delta \cdot 2} = \frac{7}{2} \\
G_c &= \frac{C - C_{nc}}{C} = \frac{1}{8}
\end{aligned}
\tag{3.12}
$$

It takes 8 transmissions to exchange two packets between two stations without network coding. Thus, we have $C = 8/2 = 4$. With network coding, however, it only takes 7 transmissions. Thus, we have $C_{nc} = 7/2$.

3.3.5 Different overlapping intervals

The above discussion is based on the assumption that buses are perfectly synchronized: they arrive at and leave the stations at the same time. However, it does not perfectly reflect how real PTN bus lines function. Therefore, next, we examine what happens when bus waiting times at the cross station overlap partially.

Figure 3.4 illustrates the main notations used to represent the partially overlapping intervals of two buses. Here, the time spent waiting at the central station, t_c, is divided into three parts: interval t_{12}, where both buses are connected to the cross station, and two evenly spaced intervals where either the bus b_1 or b_2 is connected. It is possible to derive the expressions of P, P_{nc} and G_p as a function of t_{12}. For the sake of brevity, only the case where the base station cannot drain

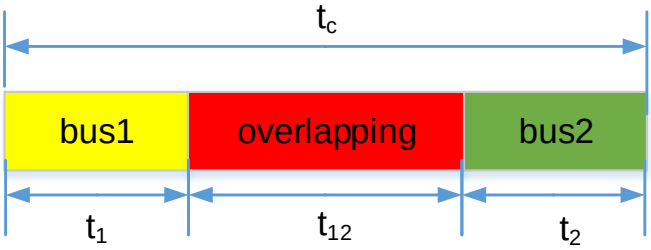


Figure 3.4 – The overlapping waiting time at the cross station

messages with network coding for the two bus lines scenario is given. Again we assume $t_1 = t_2$:

$$\begin{aligned} P &= \frac{N}{L} = \frac{\frac{t_c - t_{12}}{2} + \frac{t_{12}}{3}}{2 \cdot L} = \frac{3 \cdot t_c - t_{12}}{12 \cdot T} \cdot \Delta \\ P_{nc} &= \frac{N_{nc}}{L} = \frac{\frac{t_c - t_{12}}{2} + \frac{t_{12}}{3} \cdot 2}{2 \cdot L} = \frac{3 \cdot t_c + t_{12}}{12 \cdot T} \cdot \Delta \\ G_p &= \frac{N_{nc} - N}{N_b \cdot L} = \frac{t_{12}}{6 \cdot T} \cdot \Delta \end{aligned} \tag{3.13}$$

Clearly, G_p is a linear function of t_{12}. The more overlapping intervals, the more improvements achieved by network coding. The maximum gain, $G_p = 1/2$, is achieved by network coding when buses overlap completely, i.e. $t_c = t_{12}$. In contrast, there is no improvement if the overlapping is null, i.e. $t_{12} = 0$.

3.3.6 Performance improvement in real PTNs

In this section, using the theoretical analysis introduced in Section 3.3.5 and accurate information about the PTNs of three different cities, Toulouse, Paris and Helsinki, we show the potential for improvement when using XOR network coding in our content delivery platform.

Using the publicly available data, we calculate for all pairs of bus lines in the PTN and at each station they cross the total duration buses spend waiting alone (t_1 for line 1, t_2 for line 2) and the time they spend waiting together t_{12}. With the waiting time (t_1, t_2) and the overlapping intervals t_{12} of a pair of buses $(line1, line2)$, we can derive the amount of data that a pair of bus lines can exchange with and

without network coding using Eq. (3.14) and Eq. (3.15), respectively:

$$D_{nc} = [min(t_1, t_2) + \frac{2}{3} \cdot t_{12}] \cdot R \tag{3.14}$$

$$D = [min(t_1, t_2) + \frac{1}{3} \cdot t_{12}] \cdot R \tag{3.15}$$

where R denotes the data rate.

The expected maximum throughput gains G_t is defined as $(D_{nc} - D)/D$ and the overlapping ratio r is defined as $t_{12}/(t_1 + t_{12} + t_2)$. From these values, an upper bound on the throughput gain achievable using our pairwise inter-session XOR network coding solution can be calculated.

Table 3.2 shows a part [1] of the throughput improvements that can be achieved during week days [2] in Toulouse assuming $R = 100\ Mb/s$. Take for example the first row $(10, 2)$: the bus lines 10 and 2 cross at *Cours Dillon*. During the period from 7:00 to 19:00, their overlapping intervals are two hours and therefore the ratio of the overlapping intervals is 80.49%. The amount of data that those two lines can exchange is 163 *GB without* network coding but 309.5 *GB with* network coding, an improvement of 89.88%.

Table 3.3 shows the overall network throughput for the Toulouse, Paris and Helsinki topologies. The potential gain is significant and thus, XOR-network coding is a promising solution for improving the performance of our content delivery infrastructure.

Note that, the upper bounds shown here are calculated assuming all pairwise bus encounters take place without other buses being present at the same time at the station. Obviously, in reality this is not necessarily the case. Additional buses will bring more congestion to the station access point and reduce the benefit of pairwise network coding. The exact XOR network-coding benefit using fine-grained simulations is shown in Section 3.5.4.

1. The complete table is given in Appendix A.
2. The Toulouse PTN uses for service types: Weekday, Saturday and Sunday, Holiday, Weekday during School Holidays.

line1	**line2**	t_1	t_{12}	t_2	r(%)	D(**GB**)	D_{nc}(**GB**)	G_t(%)	**station**
10	2	0:22:00	9:46:00	2:00:00	80.49	163.0	309.5	89.88	Cours Dillon
10	109	1:19:00	8:55:00	1:40:00	74.93	193.0	326.75	69.3	Malepère
204	205	2:01:00	2:57:00	1:11:00	47.97	97.5	141.75	45.38	Ayguevives Collège
35	83	0:42:00	9:21:00	1:38:00	80.03	171.75	312.0	81.66	Balma-Gramont
35	20	0:34:00	9:29:00	1:28:00	82.34	167.75	310.0	84.8	Balma-Gramont
35	51	3:09:00	6:54:00	0:44:00	63.99	136.5	240.0	75.82	Balma-Gramont
35	77	3:34:00	6:29:00	1:24:00	56.62	160.25	257.5	60.69	Balma-Gramont
35	A	0:00:00	10:03:00	2:17:00	81.49	150.75	301.5	100.0	Balma-Gramont
2	81	2:23:00	9:15:00	0:20:00	77.3	153.75	292.5	90.24	Université Paul Sabatier
2	78	0:49:00	10:49:00	0:36:00	88.42	189.25	351.5	85.73	Université Paul Sabatier
2	56	2:07:00	9:31:00	0:22:00	79.31	159.25	302.0	89.64	Université Paul Sabatier
2	115	2:09:00	9:29:00	0:16:00	79.69	154.25	296.5	92.22	Université Paul Sabatier
2	34	2:04:00	9:34:00	0:19:00	80.06	157.75	301.25	90.97	Université Paul Sabatier
2	82	9:45:00	1:53:00	0:02:00	16.14	29.75	58.0	94.96	Université Paul Sabatier
50	11	1:24:00	9:41:00	0:38:00	82.65	173.75	319.0	83.6	Basso Cambo
50	49	0:57:00	10:08:00	0:31:00	87.36	175.25	327.25	86.73	Basso Cambo
50	53	1:22:00	9:43:00	0:51:00	81.42	184.0	329.75	79.21	Basso Cambo
50	14	0:28:00	10:37:00	1:14:00	86.2	180.25	339.5	88.35	Basso Cambo
50	48	6:23:00	4:42:00	0:23:00	40.99	87.75	158.25	80.34	Basso Cambo
50	A	0:00:00	11:05:00	1:15:00	89.86	166.25	332.5	100.0	Basso Cambo
50	57	2:16:00	8:49:00	0:56:00	73.37	174.25	306.5	75.9	Basso Cambo
50	58	1:59:00	9:06:00	0:31:00	78.45	159.75	296.25	85.45	Basso Cambo
50	8	1:53:00	9:12:00	0:56:00	76.56	180.0	318.0	76.67	Basso Cambo
81	109	0:36:00	3:09:00	6:55:00	29.53	74.25	121.5	63.64	Castanet-Tolosan
81	78	0:25:00	9:10:00	2:15:00	77.46	156.25	293.75	88.0	Université Paul Sabatier
81	56	2:14:00	7:21:00	2:32:00	60.66	210.75	321.0	52.31	Université Paul Sabatier
81	115	1:32:00	8:03:00	1:42:00	71.34	189.75	310.5	63.64	Université Paul Sabatier
81	34	1:19:00	8:16:00	1:37:00	73.81	183.25	307.25	67.67	Université Paul Sabatier
81	202	0:35:00	3:10:00	7:36:00	27.9	73.75	121.25	64.41	Castanet-Tolosan
81	62	0:20:00	3:25:00	7:58:00	29.16	66.25	117.5	77.36	Castanet-Tolosan
81	82	8:19:00	1:16:00	0:39:00	12.38	48.25	67.25	39.38	Université Paul Sabatier
81	205	0:37:00	3:08:00	7:30:00	27.85	74.75	121.75	62.88	Castanet-Tolosan
83	201	1:06:00	9:52:00	0:40:00	84.81	178.0	326.0	83.15	Saint Orens Lycée
83	79	0:45:00	10:13:00	1:08:00	84.44	187.0	340.25	81.95	Saint Orens Lycée
83	78	0:02:00	10:56:00	1:11:00	89.99	165.5	329.5	99.09	Saint Orens Lycée
83	20	0:47:00	10:12:00	0:45:00	86.93	186.75	339.75	81.93	Balma-Gramont
83	51	3:38:00	7:21:00	0:17:00	65.24	123.0	233.25	89.63	Balma-Gramont
83	77	3:42:00	7:17:00	0:36:00	62.88	136.25	245.5	80.18	Balma-Gramont
83	A	0:00:00	10:59:00	1:21:00	89.05	164.75	329.5	100.0	Balma-Gramont
73	B	1:38:00	9:32:00	0:49:00	79.55	179.75	322.75	79.55	Borderouge
73	26	1:12:00	9:58:00	0:46:00	83.52	184.0	333.5	81.25	Borderouge
...	...	...	...	...	...	...	...	...	...
In total:		-	-	-	71.58	22325.25	40762.5	82.58	-

Table 3.2 – The throughput improvements achieved by network coding in Toulouse

City	r(%)	D(TB)	D_{nc}(TB)	G_t(%)
Toulouse	71.58	22.33	40.76	82.58
Paris	71.43	107.26	193.71	80.59
Helsinki	58.0	412.01	721.04	75.01

Table 3.3 – The potential improvements: upper bound

3.4 XOR network coding implementation for PTNs

This section presents the implementation of XOR network coding for our content delivery infrastructure. Encoding and decoding operations are only performed at the stations. Buses are carrying coded messages that are decoded at the next hop station. Thus, they don't store any previously carried messages. This feature is important as a bus may not possess the message necessary to decode XOR-ed ones by simply keeping the history of previously carried messages. This is typically the case if previous messages were carried by a different bus of the same bus line.

If network coding is performed at a station, the virtual message queue Q_j defined for the messages routed to next hop s_j is further divided into several network coding queues. These network coding queues are indexed by a 2-tuple key (s_i, s_j) where s_i and s_j denote the previous hop and the next hop identifier of a message. A network coding queue is referred to using notation Q_{ij}.

For instance on Figure 3.2, $P1$ is stored in queue Q_{12} and $P2$ in queue Q_{21}. Encoding and decoding algorithms are introduced next. All network coding operations are limited to the 1-hop neighborhood of the station s where buses enter in contact.

3.4.1 Encoding procedure

The pseudo-code of encoding procedure is listed in Algorithm 3.1. We assume that at the time of encoding, a station s has a list of buses currently waiting at the station. It can easily obtain a list of the next hop stations S that are reachable with all buses currently waiting.

The station goes through S to find two non-empty message queues: Q_{ij} and Q_{ji} $(i \neq j)$ [1]. With this selection, two local cross communications are identified that can directly benefit from network coding. Next, the two head-of-line messages m_i and m_j are picked from Q_{ij} and Q_{ji} respectively. A new message m_c is created by xor-ing m_i and m_j together (i.e. $m_c = m_i \oplus m_j$). Station s broadcasts m_c in a single transmission. With this selection, we ensure that m_c can be decoded at both next-hop stations s_i and s_j.

1. Complexity of getting these two non-empty queues is of $|S|^2/4$ on average.

If no cross communication is found, basic unidirectional forwarding operations are performed to reduce delays.

Algorithm 3.1 Coding procedure

```
B = {Buses waiting at the station }
S = {Next hop nodes to be reached by B }
for all s_i ∈ S do
    for all s_j ∈ S, j ≠ i do
        if Q_ij ≠ ∅ and Q_ji ≠ ∅ then
            m_i is picked at the head of Q_ij
            m_j is picked at the head of Q_ji

            return m_c = m_i ⊕ m_j
        end if
    end for
end for
```

3.4.2 Decoding procedure

Each station keeps the messages that it has given to buses. The messages are stored in a hash table keyed by its message identifier. When a station receives an XOR-ed message $m_c = m_i \oplus m_j$, it looks through the hash table to get the previously sent message, say m_i. A new message m_j is retrieved from m_c by XOR-ing it with m_i, i.e. $m_j = m_c \oplus m_i$. Once the message m_j is decoded, it is stored into the virtual network coding queue. Message m_i is discarded. The previous station and the next station of m_j are extracted to store it in the appropriate virtual network coding queue.

3.5 Performance evaluation

In this section, we use trace-based simulations to evaluate the performance of the proposed XOR network coding strategy for our content delivery infrastructure.

Table 3.4 – Model and simulation parameters

Simulation Parameter	Value	Model Parameter	Value
t_l	100s	simulation duration	43200s
t_b	100s	time-to-live	6000s
t_c	100s	buffer size	infinite
N_b	2, 4	-	-

3.5.1 Simulation setup

We use the ONE simulator [70] and add the implementation of network coding described in Section 3.4.

The XOR network coding strategy is compared to the **Baseline** strategy where all nodes are equipped with a wireless AP and simply forward packets. The strategy where all nodes perform network coding is named **ALL-NC**.

3.5.2 Evaluation for two bus lines

The delivery probability

Figure 3.5 shows both theoretical and simulation-based delivery probabilities. According to Section 3.3.2 and with the parameters of Table 3.4, the theoretical delivery probability without network coding P, with network coding P_{nc} and the gain G_p are calculated for different message creation periods Δ using Eq. (3.16). Clearly, the simulation results are consistent with the theoretical ones and in this case the maximum gain of 48% is achieved at $\Delta = 12$. They do not completely match because the simulation is done in finite time and the model assumes infinite communication in time.

$$\begin{cases} P = \frac{1}{24} \cdot \Delta, & P_{nc} = \frac{1}{12} \cdot \Delta, & G_p = \frac{1}{24} \cdot \Delta, & \Delta < 12 \\ P = \frac{1}{8} \cdot \Delta - 1, & P_{nc} = 1, & G_p = 2 - \frac{1}{8} \cdot \Delta, & 12 \leq \Delta < 16 \\ P = 1, & P_{nc} = 1, & G_p = 0, & \Delta \geq 16 \end{cases} \tag{3.16}$$

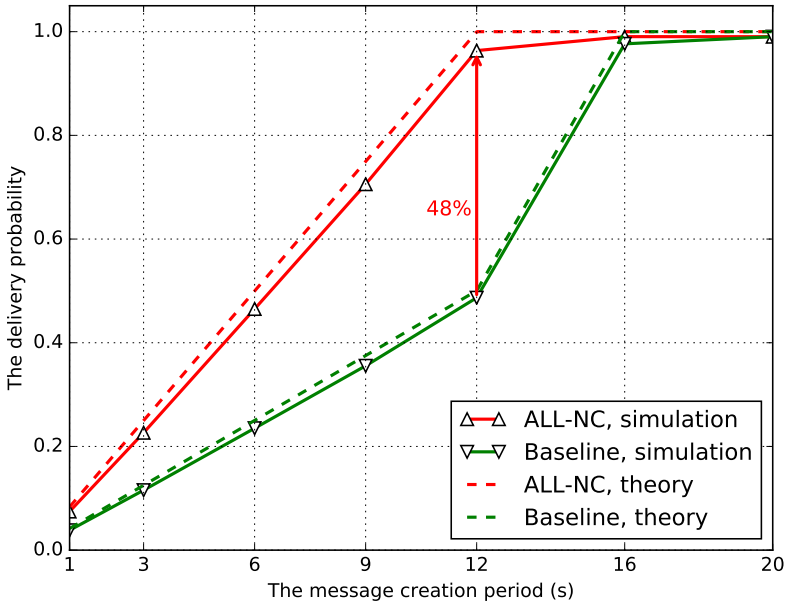


Figure 3.5 – Delivery probability for $N_b = 2$

The overhead ratio

According to Section 3.3.4 and with the parameters of Table 3.4, the overhead ratio without network coding C, with network coding C_{nc} and the gain G_c are calculated for different message creation periods Δ using Eq. (3.17).

Figure 3.6 shows the clear benefit of network coding when it comes to reducing the overhead ratio. At high loads, the cross station is able to drain twice as much data per transmission opportunity when it uses network coding. The simulation result is slightly higher than the theoretical one. This is because *i*) the encoding probability at the cross station is less than 100% thanks to the random access channel; *ii*) the simulation is done in finite time and at the end of simulation some messages are still on the buses and the cross station.

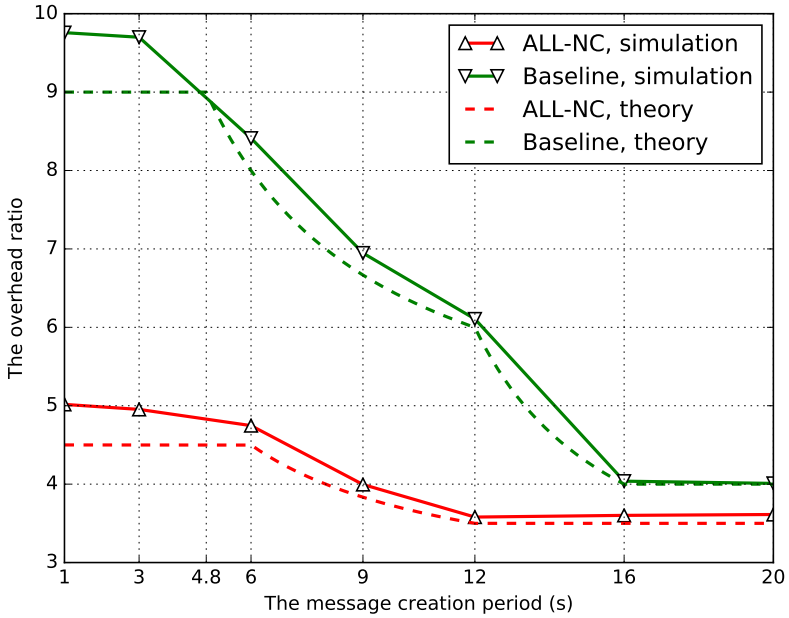


Figure 3.6 – The overhead ratio for $N_b = 2$

$$\begin{cases} C = 9, & C_{nc} = \frac{9}{2}, & G_c = \frac{1}{2}, & \Delta < \frac{24}{5} \\ C = 4 + \frac{24}{\Delta}, & C_{nc} = \frac{9}{2}, & G_c = \frac{48 - \Delta}{48 + 8 \cdot \Delta}, & \frac{24}{5} \leq \Delta < 6 \\ C = 4 + \frac{24}{\Delta}, & C_{nc} = \frac{5}{2} + \frac{12}{\Delta}, & G_c = \frac{24 + 3 \cdot \Delta}{48 + 8 \cdot \Delta}, & 6 \leq \Delta < 12 \\ C = \frac{2 \cdot \Delta}{\Delta - 8}, & C_{nc} = \frac{7}{2}, & G_c = \frac{56 - 3 \cdot \Delta}{4 \cdot \Delta}, & 12 \leq \Delta < 16 \\ C = 4, & C_{nc} = \frac{7}{2}, & G_c = \frac{1}{8}, & \Delta \geq 16 \end{cases} \tag{3.17}$$

The average latency

Figure 3.7 shows that the average latency reduces with network coding using simulations. This is because when performing network coding, messages at the cross station can be drained twice as fast as without network coding and therefore messages arrive at their destinations much earlier.

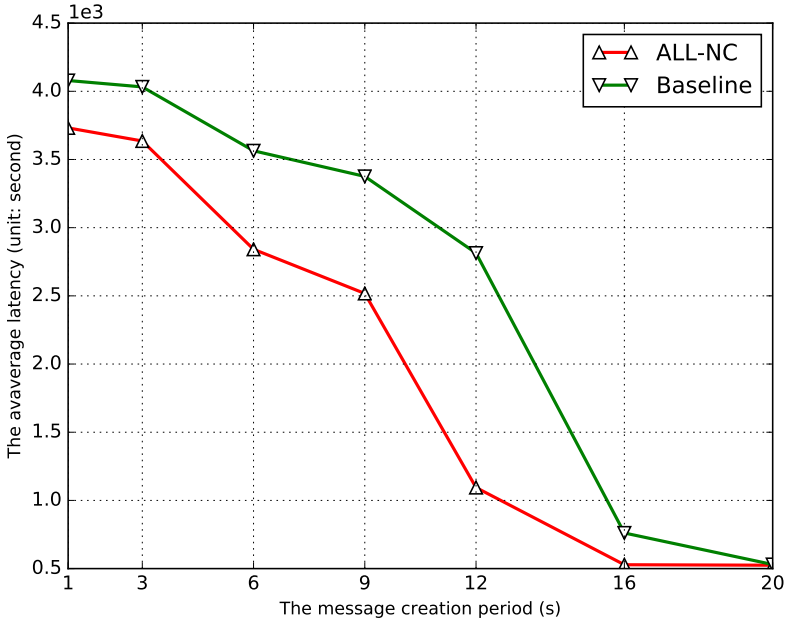


Figure 3.7 – The average latency for $N_b = 2$

3.5.3 Effect of the overlapping interval

To be consistent with the theoretical analysis of Section 3.3.5, we set the message creation interval Δ to 12 to explore via simulations how the overlapping interval t_{12} affects the delivery probability. With the parameters of Table 3.4, we calculate the delivery probability without network coding P, with network coding P_{nc} and the gain G_p for different overlapping intervals t_{12} as follows:

$$\begin{cases} P = \dfrac{300 - t_{12}}{400} \\ P_{nc} = \dfrac{300 + t_{12}}{400} \\ G_p = \dfrac{t_{12}}{200} \end{cases} \tag{3.18}$$

As Figure 3.8 shows, not surprisingly, the increase of the overlapping interval increases the delivery probability of network coding while it has the opposite effect on Baseline. As the overlapping interval increases, the delivery probability without network coding drops rapidly due to the load/bandwidth imbalance between the cross station and the buses. On the other hand, as the overlapping increases there

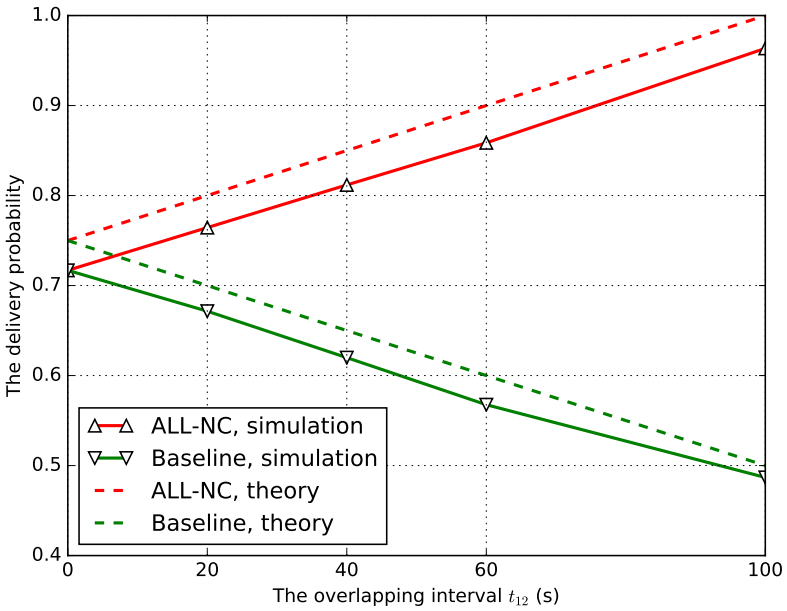


Figure 3.8 – The delivery probability as a function of the overlapping interval ($N_b = 2$).

are more opportunities for network coding, allowing the cross station to drain its packets almost twice as fast, explaining the observed improvements. It is easy to understand that there is no improvement at $t_{12} = 0$. In contrast, the maximum gain is achieved at $t_{12} = 100$ when we get complete overlapping.

Similarly, according to the theoretical analysis of Section 3.3.3 and with the parameters of Table 3.4, we are able to calculate the delivery probability without network coding P, with network coding P_{nc} and the gain G_p for different message creation periods Δ. The theoretical and simulation-based results for $N_b = 4$ are presented in Eq. (3.19) and Figure 3.9 respectively. The maximum gain is still around 50% when $\Delta = 24$. Compared to the two-bus-line scenario, the maximum gain needs twice as long to reach 50% for the reason that the cross station has twice as many packets to drain because of the presence of two cross flows.

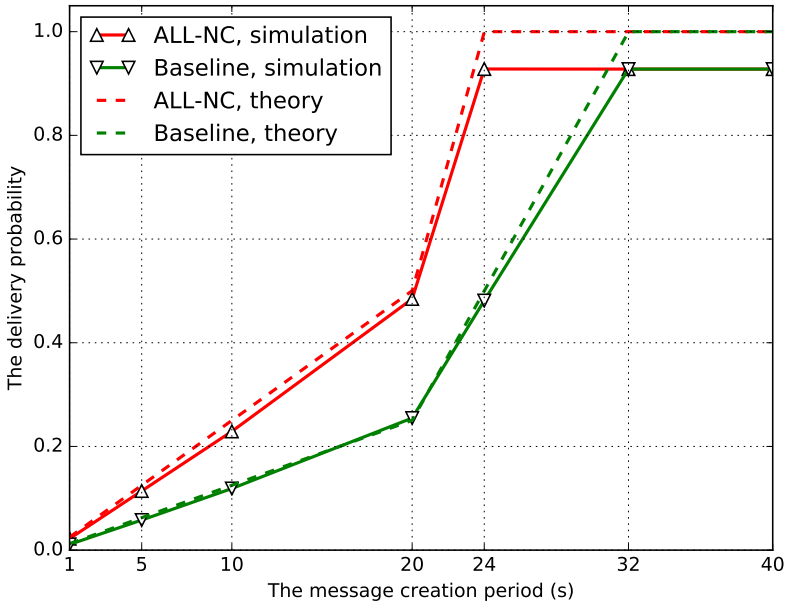


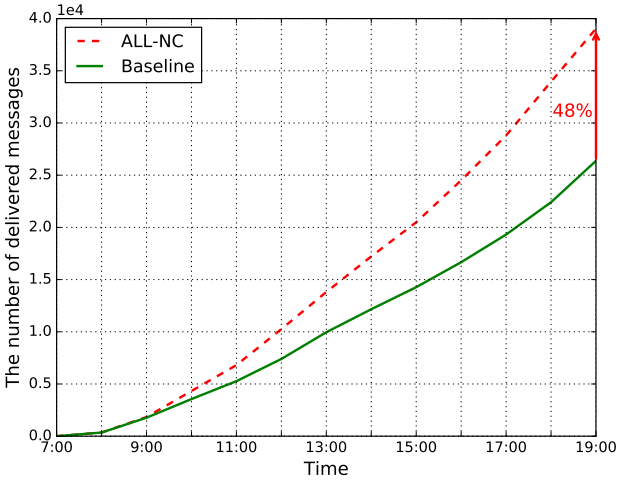
Figure 3.9 – The delivery probability for $N_b = 4$

$$\begin{cases} P = \frac{1}{80} \cdot \Delta, & P_{nc} = \frac{1}{40} \cdot \Delta, & G_p = \frac{1}{80} \cdot \Delta, & \Delta \leq 20 \\ P = \frac{1}{16} \cdot \Delta - 1, & P_{nc} = \frac{1}{8} \cdot \Delta - 2, & G_p = \frac{1}{16} \cdot \Delta, & 20 < \Delta < 24 \\ P = \frac{1}{16} \cdot \Delta - 1, & P_{nc} = 1, & G_p = 2 - \frac{1}{16} \cdot \Delta, & 24 \leq \Delta < 32 \\ P = 1, & P_{nc} = 1, & G_p = 0, & \Delta \geq 32 \end{cases} \tag{3.19}$$

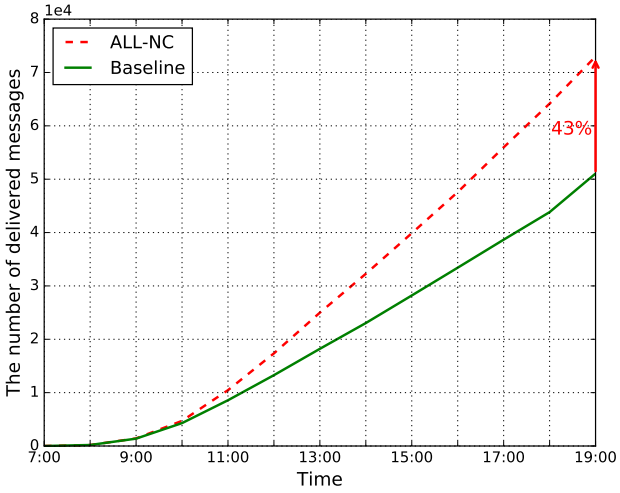
3.5.4 Evaluation of network coding benefits in real PTNs

In this section, we aim at evaluating the benefit that can be brought if all PTN stations perform the XOR network coding strategy of Algorithm 3.1. Extensive simulations are carried out using the simulation setup described in Section 2.5.1. Two metrics are evaluated: *i*) the number of delivered messages and *ii*) the overhead ratio. The overhead ratio is defined as the ratio of the number of times any message was transferred at any station to the number of messages delivered.

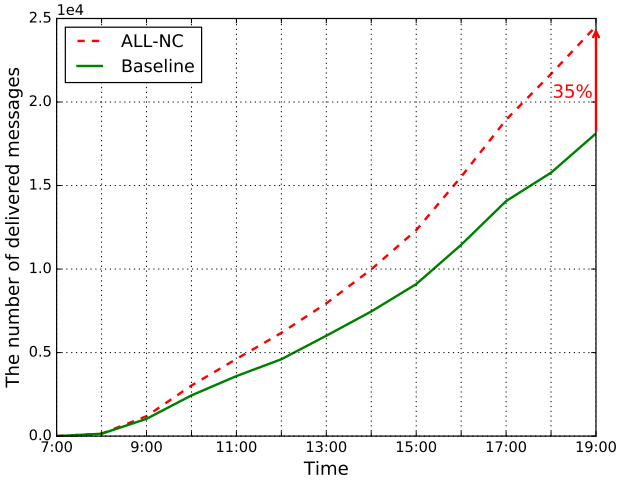
Figure 3.10 shows the evolution of the number of delivered messages for baseline



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 3.10 – Network coding benefit : number of delivered messages.

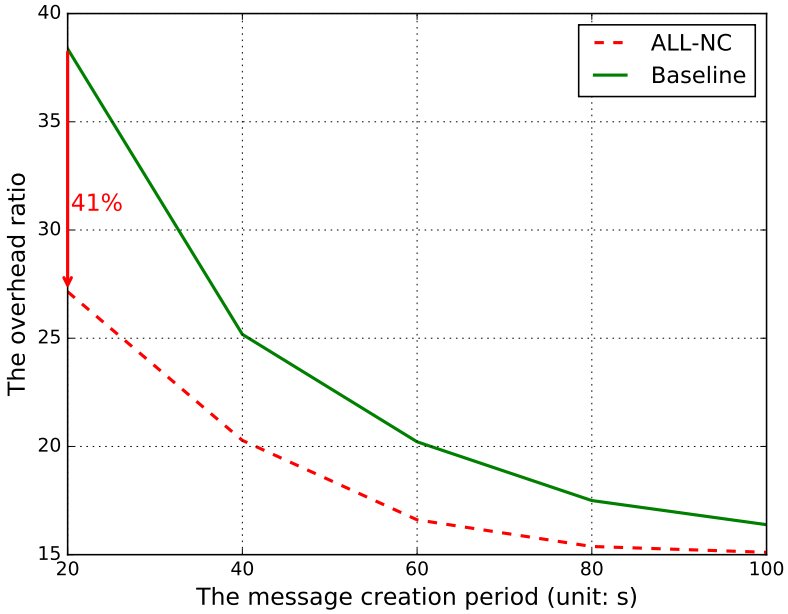


Figure 3.11 – The overhead ratio in Paris.

and ALL-NC over a period starting at 7:00 a.m. and ending at 7:00 p.m. Clearly, ALL-NC outperforms the baseline strategy as expected. Improvements reach 48%, 43% and 35.5% in Toulouse, Paris and Helsinki, respectively. These gains are significant but not as large as the upper bound presented in Table 3.3 (that are respectively of 82.6%, 80.6% and 75%). As already explained, this upper bound is optimistic since *i*) the pair-wise network coding cannot totally compensate the traffic imbalance if more than two buses are in contact at the station; *ii*) at the end of simulation, a considerable amount of messages are still at the buses and the cross stations.

This gain could be improved if the station could XOR more than two packets together as done in the wheel topology of [54]. But this is unfortunately not possible as the 1-hop decoding stations cannot overhear the remote message emissions of the buses as done in a pure wireless setting.

Nevertheless, the benefit of ALL-NC is really significant: messages get delivered much faster with network coding since stations are capable of draining twice as many packets as without network coding. The number of delivered messages obtained with network coding reach the same level as the one obtained for non network coding about 2.5 hours earlier.

The overhead ratio is also reduced with network coding, as shown in Figure 3.11. In this plot, the message creation period Δ is increased to reduce traffic. The overhead is nicely reduced since two stations can extract desired messages with XOR network coding in a single transmission instead of two with baseline. In the Paris topology, the decrease reaches 41% which is captured at $\Delta = 20$. With Δ increasing, the gap between baseline and ALL-NC is narrowed since traffic is reduced and thus, coding opportunities become less frequent.

3.6 Conclusion

In this chapter, we introduced the idea of leveraging inter-session XOR hop-by-hop network coding to address the congestion arising from many bus lines converging in few major stops in public transportation networks (PTNs). We verified the benefits of our network coding scheme in various scenarios through both theoretical analysis and simulations. Our results showed that the maximum delivery probability is increased by 50% while the maximum overhead ratio is decreased by 50%. Simulations using real PTN traces from three European cities showed that a network-coding based content delivery infrastructure can improve message delivery between 35% - 48% over a system using the traditional communication model.

4 Towards a Cost-effective Design

In Chapter 3, we have demonstrated the benefits of introducing XOR network coding in a delay tolerant network created by a public transportation network. However, a straightforward rollout of this content delivery infrastructure requires the deployment of a large number of APs. Our aim in this chapter is to create a network architecture that provides the same performance as the one of Chapter 3 but in a cost-efficient manner.

4.1 Introduction

A natural solution to this problem is to reduce the number of APs rolled out in the city by introducing a more advanced routing strategy as described in Section 4.2. In this solution, APs will be only deployed at strategic locations forming a connected dominating set (CDS). Buses will carry their messages until reaching the AP that is part of the CDS. Only the buses that connect CDS nodes will route the messages using a pre-calculated shortest path to the destination AP.

We show that the state-of-the-art minimum connected dominating set calculation doesn't offer the same offloading performance than the ALL-AP solution of Chapter 2. A dedicated CDS creation heuristic is then proposed to favor the selection of end-stations where network coding is the most beneficial for offloading performance. We will refer to this new design as the 2-tier architecture. It groups bus end-stations into two groups (named tiers here): *i*) the end-stations equipped with APs composes the first tier; *ii*) the end-stations where no APs are rolled out composes the second tier.

Section 4.3 investigates a further cost improvement that is sustained by the

following observation. A network coding enabled AP would require dedicated hardware to support the advanced processing of XOR operations at high speed. Indeed, advanced queue management has to be implemented in such APs, together with proper network coding / decoding processes introduced in the previous chapter. Moreover, network coding is well-known for its vulnerability to pollution attacks. Preventing such attacks relies on advanced homomorphic cryptography which adds to the overall complexity burden.

As such, Section 4.3 investigates the idea of further dividing the first tier of the 2-tier architecture into a set of network coding enabled APs and a set of regular wireless APs. The potential of creating this 3-tier architecture is underlined in this part where we look at different AP selection criteria to extract the network coding enabled ones from the CDS.

The core notation used in this chapter to refer to the different groups of bus end-stations is the following:

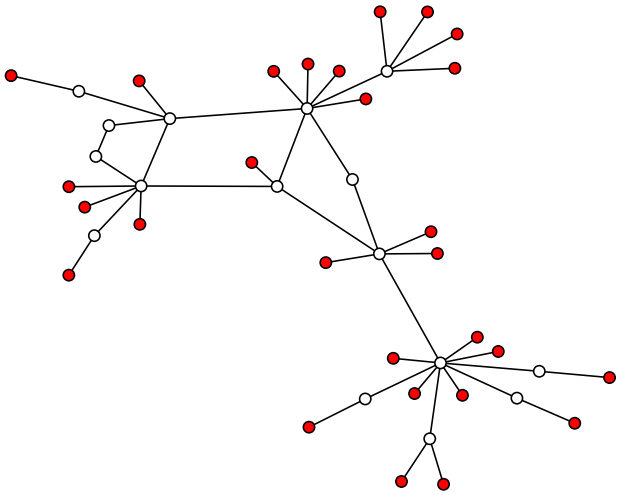
1. V denotes the full set of bus end-stations of $G(V, E)$,
2. $V' \in V$ denotes the set of end-stations equipped with basic APs,
3. $V'' \in V'$ denotes the set of end-stations with network coding enabled APs.

As such, the 2-tier architecture of section 4.2 is composed of a first tier equal to V', and a second tier equal to $V - V'$. And the 3-tier architecture of 4.3 is composed of a first tier equal to V'', a second tier equal to $V' - V''$ and a last tier equal to $V - V'$. Table 4.1 lists all notations and symbols of Chapter 4.

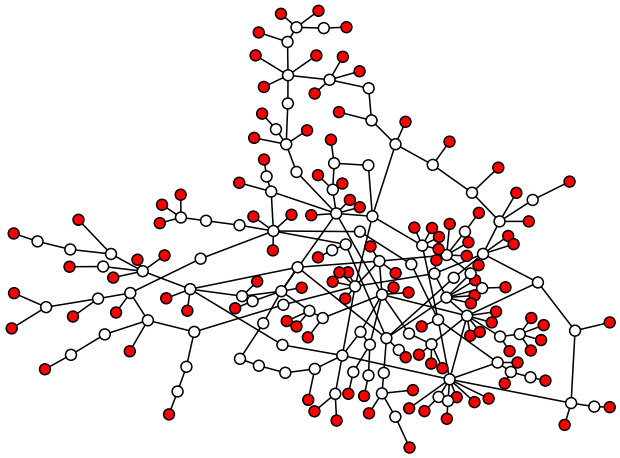
Simulation results demonstrate the viability of our design choices in 4.4. In particular, the 3-Tier architecture is shown to guarantee end-to-end connectivity and reduce the deployment cost by a factor of 3 while delivering 30% more packets than a baseline architecture. For the Paris topology, it can offload mobile data up to 4.7 terabytes [1] within 12 hours.

To summarize, the overarching goal of this chapter is to define an architecture that reduces the number of APs and network coding enabled APs such as to offer the same performance as the basic rollout of Chapter 3 given by the *ALL-NC* architecture.

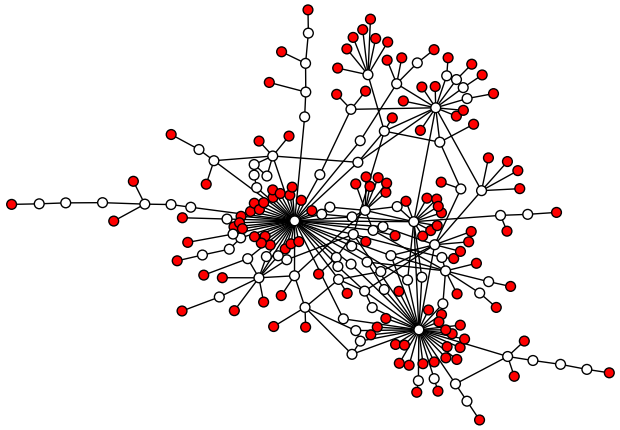
1. 1 terabyte (TB) = 1000 gigabytes (GB).



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.1 – The leaf nodes in three PTNs (in red).

Term	Description
$G(V, E)$	graph modeled from PTN
V'	a connected dominating set of G
V''	a subset of V'
$G(V', E')$	a subgraph of G induced by V'
$N(v)$	the neighbors of node v
Δ	message creation period, $\Delta > 0$
θ	threshold value on node degree
$deg(v)$	the degree of node v

Table 4.1 – Notations and symbols.

4.2 Router nodes selection

To provide a cost-efficient design for our content delivery infrastructure, a reduced number of wireless access points has to be rolled out on the premise that the end-to-end connectivity is guaranteed. Note that leaf stations ($deg(v) = 1$) do not need to relay any message because a leaf station only serves a single bus line and no data flows cross at them. Thus, there is no need for deploying router nodes at such stations. Further, maintaining full connectivity with a reduced number of routing enabled nodes is possible by selecting the ones forming a connected dominating set (CDS) of the graph.

In this Section, we recall first the definition of a connected dominating set and introduce the state-of-the-art heuristic to calculate a *minimum* connected dominating set (MCDS). Next, we explain how we leverage the MCDS to route messages in this new architecture. We demonstrate by simulation that the MCDS may negatively impact the forwarding performance in some PTN topologies compared to the ALL-NC architecture. After explaining the reason for that, we design a novel CDS heuristic (TH-CDS) that selects the nodes such as to better benefit from network coding operations and to reduce CDS calculation complexity as well. In this heuristic, we relax the minimum size constraint.

We refer to reader to Section 4.4 to get a comprehensive simulation study that underlines the improvements of TH-CDS calculation compared to MCDS on Paris, Helsinki and Toulouse topologies. It is important to notice that even though the minimum size objective is relaxed, TH-CDS produces only a slightly larger CDS

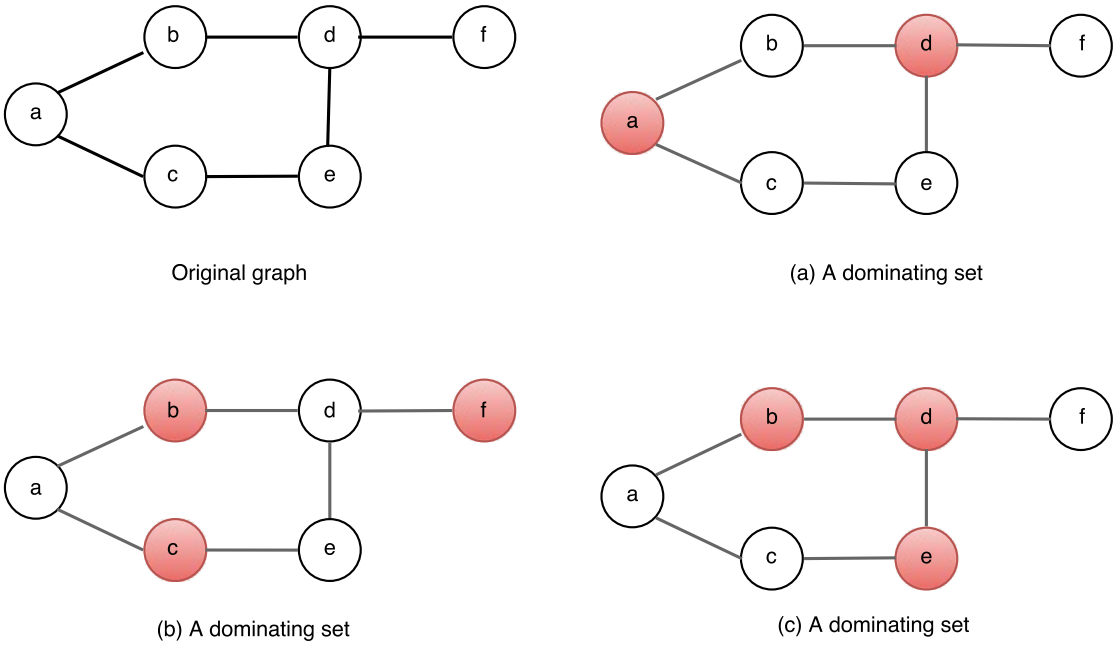


Figure 4.2 – An illustration of dominating sets for a graph, highlighted in red.

than MCDS for the considered topologies.

4.2.1 Connected dominating sets

A dominating set for a graph $G(V, E)$ is a subset D of V such that every vertex not in D is adjacent to at least one member of V. In other words, every node in $\{V - D\}$ is *dominated* by at least one of node of D. Figure 4.2 illustrates three possible dominating sets for a graph. Take Figure 4.2-(b) as an example, a, d, e is dominated by $\{b, c\}$, $\{b, f\}$ and c respectively.

A connected dominating set (CDS) [71] of a graph $G(V, E)$ is a set V' of vertices with two properties: *i*) V' is a dominating set of G; *ii*) V' induces a connected subgraph G' of G. As illustrated in Figure 4.2-(c), the red nodes form a connected dominating set since $\{b, d, e\}$ is a dominating set and its induced subgraph $(b{-}d{-}e)$ is connected.

A minimum connected dominating set (MCDS) of a graph G is a CDS with the smallest number of vertices among all connected dominating sets of G. As selecting the minimum subset of nodes forming a connected dominating set cannot

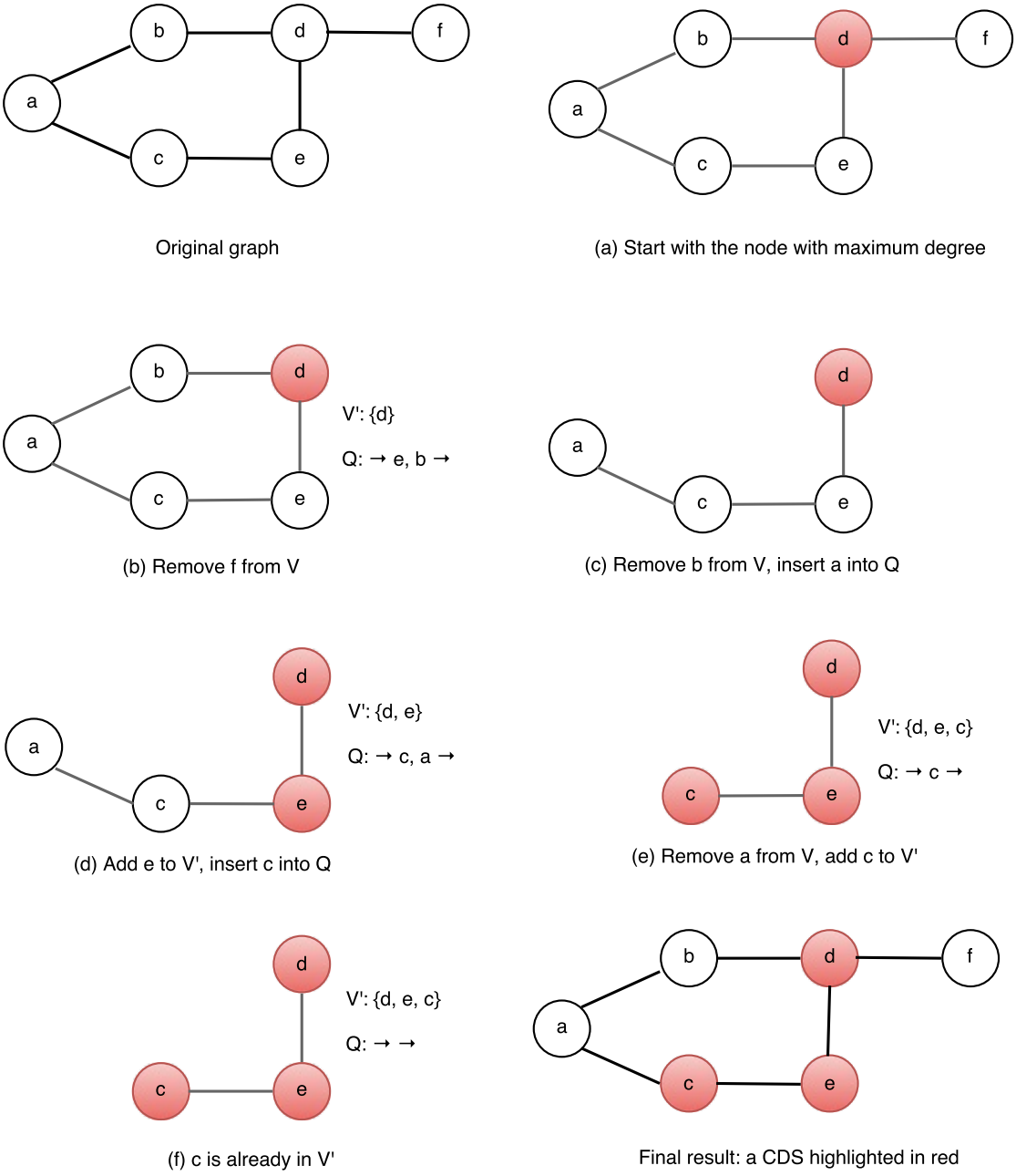


Figure 4.3 – An illustration of the procedures to construct a CDS using [1].

be solved in polynomial time [72], several approximation algorithms have been proposed to calculate an approximated minimum connected dominating set for a graph [73, 74]. Most of these algorithms fall into the following categories: *i*) Create a CDS incrementally; *ii*) Remove recursively vertices from an initial set to form a CDS; *iii*) Find a maximum independent set and then expand to a CDS by adding necessary vertices. However, these construction methods require a remarkable number of steps involved in initialization, addition/removal of vertices [1]. To address these limits, *M. Rai et al.* proposed a new heuristic approach to calculate an approximated minimum CDS [1].

Algorithm 4.1 MCDS-NON-DISTRIBUTED in [1]

Require: A connected graph $G(V, E)$
  $u \leftarrow$ the node with the maximum degree
  $V' \leftarrow \{u\}$, initialize a connected dominating set V' with u
  $Q \leftarrow N(u)$, insert all the neighbors of u into a queue Q on ascending order of degree
  while $Q \neq \varnothing$ **do**
    $v \leftarrow Dequeue(Q)$, pop a node from Q
    if $v \notin V'$ **then**
      $N^*(v) \leftarrow Sort(N(v))$, sort the neighbors of v by ascending order of degree
      if $G(V - \{v\})$ is connected **then**
        if $N(v) \cap V' = \varnothing$ **then**
          $V' \leftarrow V' \cup MAX(N^*(v))$, add the neighbor of v with maximum degree to V'
        end if
        $V \leftarrow V - \{v\}$, remove v from V
      else
        $V' \leftarrow V' \cup \{v\}$, add v to V'
      end if
      Insert the elements of $N^*(v)$ into Q only if they are not inserted and not in V'
    end if
  end while
  return V'

The main idea of [1] is to form a connected dominating set by traversing all nodes using breadth-first search, beginning with the node with the highest degree, and continuously removing the node v if $G(V - \{v\})$ is still connected. Before

diving into the details of the algorithm, we use an example to illustrate the steps to construct a CDS for a graph, as shown in Figure 4.3. A detailed description is given below.

(a) Start with the node with the maximum degree, i.e. $V' = \{d\}$. Enqueue all neighbors of d to a queue Q in ascending order of degree, say $Q = \{e, b, f\}$ (the head of the queue is on the right).

(b) Pop a node from Q, i.e. f. It can be removed from V as $G(V - \{f\})$ is still connected. f's neighbor d is already in V' and no need to be enqueued to Q.

(c) Pop a node from Q, i.e. b. b can be removed from V as b is dominated by d and the graph remains connected after removing it. b's neighbor a is inserted into $Q = \{a, e\}$.

(d) Pop a node from Q, i.e. e. e must be added to V' as $G(V - \{e\})$ is disconnected. e's neighbor c is inserted into $Q = \{c, a\}$.

(e) Pop a node from Q, i.e. a. a can be removed from V as $G(V - \{a\})$ is connected, but a's neighbor c must be added to V' to make sure a is dominated.

(f) Pop a node from Q, i.e. c. Continue the loop as c is in V'. The program stops as Q is empty.

Finally, we get a connected dominating set $V' = \{d, e, c\}$ where $\{a, b, f\}$ is dominated by c, d and d respectively and the subgraph $(c - e - d)$ induced by V' is connected, as shown in the last subfigure in Figure 4.3.

Formally, the pseudo-code is listed in Algorithm 4.1, including the following steps:

1. Take the node with the maximum degree (say u) as a starting node, i.e. initialize a connected dominating set V' with u.
2. Insert all neighbors of u (denoted by $N(u)$) that do not belong to V' (i.e. $u \notin V'$) into a queue Q sorted by ascending order of degree (the lowest degree node is at the head of Q).
3. Pop a node (say v) from Q. If $G(V - \{v\})$ is disconnected, v must be a part of CDS; otherwise, it can be excluded from CDS if v has been already dominated by one of V' and if not, add the neighbor of v with the maximum

degree to V'. Finally, push all the neighbors of v into Q only if they neither belong to Q nor to V'.

4. Do step 3 until Q is empty, i.e. traverse all nodes in G using breadth-first search.
5. In the end, V' forms a connected dominating set.

We will refer from now on to the CDS obtained with Algorithm 4.1 using the MCDS acronym.

4.2.2 2-Tier selection

It is possible to route all data traffic in a network only with CDS nodes. A connected dominating set V' is in charge of routing data to the destination [75, 76]. In this case, all non-CDS nodes push data to their dominating node. The data goes through the subgraph G' induced by V' and is routed to the destination. For instance, in Figure 4.4, a wants to send a message to b. The message is first sent to c that dominates a, then relayed to d through the subgraph $(c-e-d)$ induced by the CDS ($V' = \{d, e, c\}$), and finally delivered to the destination b.

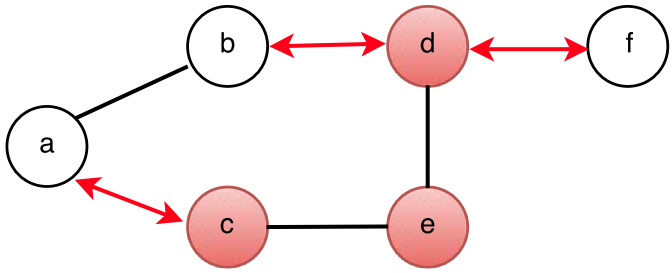


Figure 4.4 – An illustration of routing data in 2-Tier

As for our infrastructure, only the stations belonging to CDS are routing data between bus lines. As such, we only deploy wireless APs at the stations of CDS and no AP is installed at remaining stations $\{V - V'\}$. We refer to such a scheme as **2-Tier**. Fixed shortest paths (in number of hops) are calculated to route all messages within the CDS.

The MCDS of Algorithm 4.1 has been calculated for the three PTNs of Toulouse, Helsinki and Paris. The resulting topologies are plotted in Figure 4.5 where the

City	Baseline	ALL-NC	2-Tier with MCDS
Toulouse	44	44	13
Paris	213	213	85
Helsinki	217	217	60

Table 4.2 – Number of wireless access points required to cover three different cities. The 2-Tier architecture reduces the required number of interfaces by approximately a factor of 3 using MCDS.

CDS nodes are highlighted in red. Table 4.2 shows the number of wireless access points necessary to cover three major cities using the Baseline, ALL-NC and 2-Tier architectures. The results show that 2-Tier is able to reduce the number of APs by approximately a factor of 3.

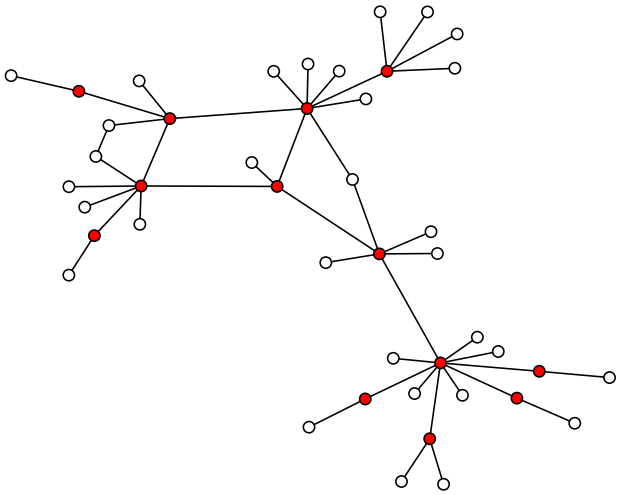
Algorithm 4.2 Calculate the route between two non-CDS nodes s, d

V' is a CDS of $G(V, E)$. V' induces a subgraph G' of G.
$S \leftarrow N(s) \cap V'$ ▷ Get the nodes that dominate s
$D \leftarrow N(d) \cap V'$ ▷ Get the nodes that dominate d
$path \leftarrow \min_{\forall s' \in S, \forall d' \in D} shortest\_path(G', s', d')$

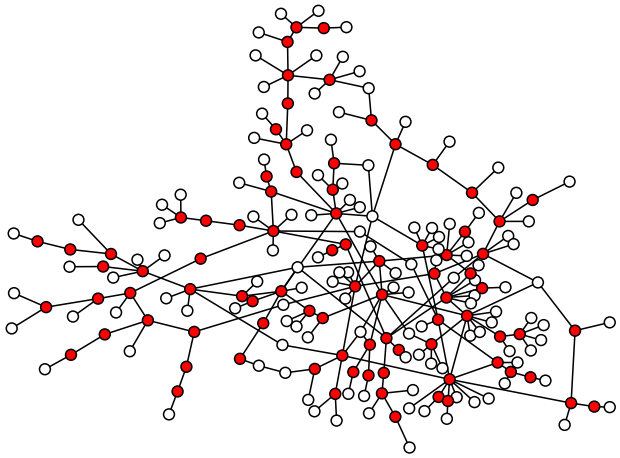
return $s \rightarrow path \rightarrow d$

The routing policy. We have explained in detail the routing policy of ALL-NC in Chapter 3. The routing policy of 2-Tier is essentially the same as that of ALL-NC, except for the routing table. In ALL-NC, we pre-calculate routing tables for each station in the whole graph G as all stations are equipped with network coding enabled APs and all of them can be router nodes. In 2-Tier, as pinpointed above, only CDS stations are equipped with network coding enabled APs and non-CDS stations push data to their dominated station.

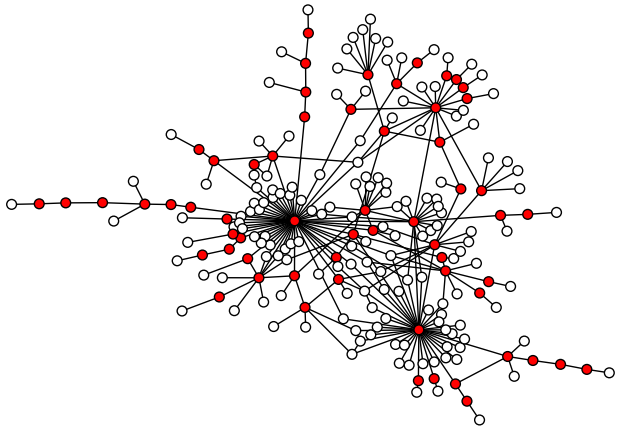
The path between s and d in a graph $G(V, E)$ simply equal to their shortest path in G' if both s and d are in V'. If not, the shortest path is calculated between their relative dominating nodes. Algorithm 4.2 presents how to calculate the route between two non-CDS nodes.



(a) Toulouse

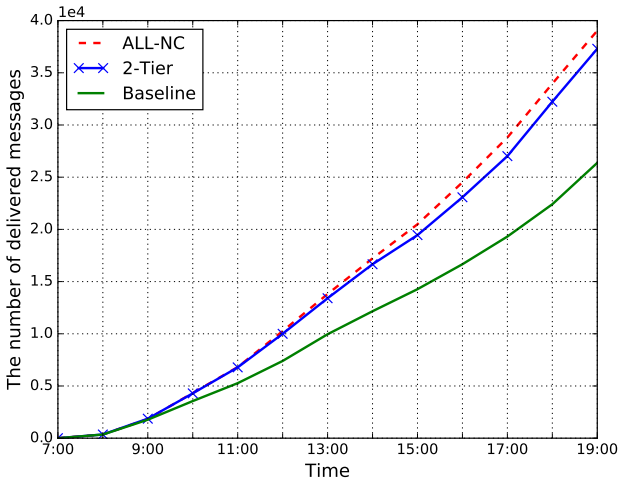


(b) Paris

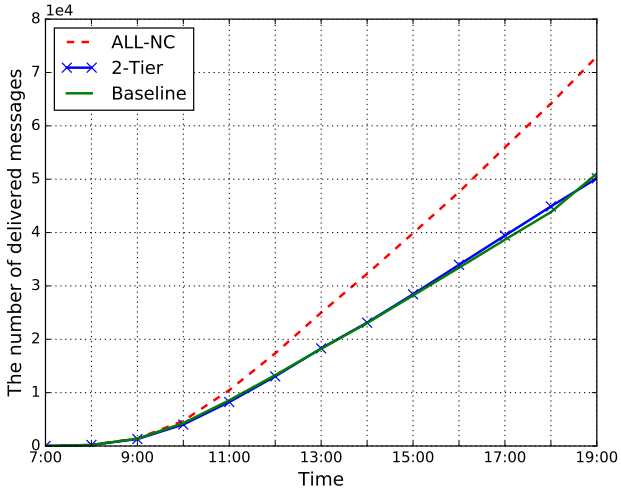


(c) Helsinki

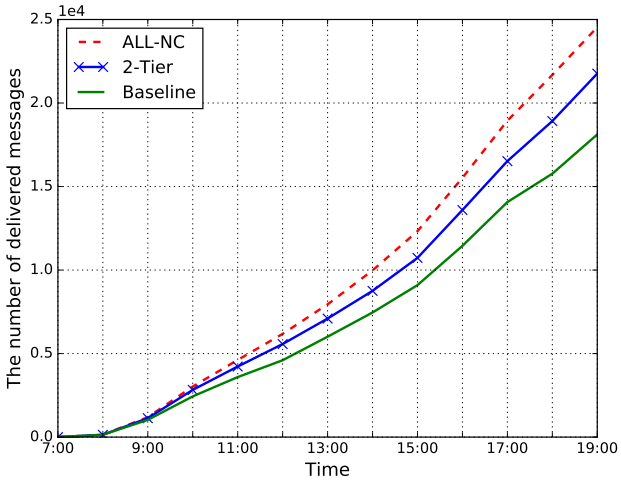
Figure 4.5 – A CDS (in red) calculated with Algorithm 4.1 in three PTNs.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.6 – Number of messages delivered for Baseline, ALL-NC and 2-Tier.

4.2.3 Evaluation of 2-Tier

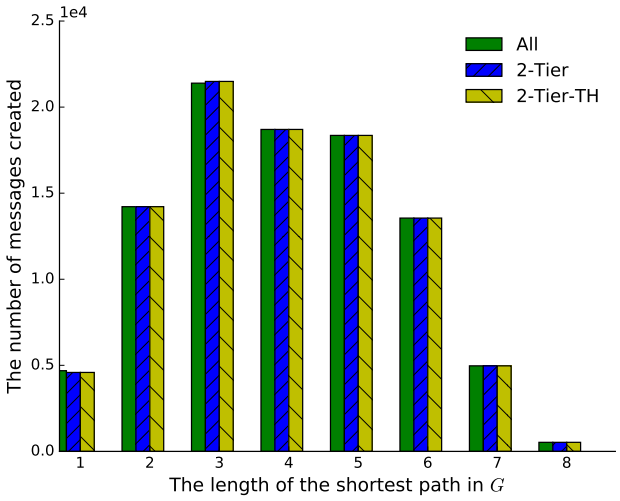
We have shown in Section 4.2.2 that the 2-Tier architecture can reduce the number of wireless access points by approximately a factor of 3 in our content delivery infrastructure. In this subsection, we evaluate the performance of 2-Tier in terms of packet delivery. More specifically, we compare the number of delivered messages of the 2-Tier architecture with **Baseline** introduced in Chapter 2 and **ALL-NC** introduced in Chapter 3. The simulation is carried out using the same simulation setup described in Section 2.5.1.

The simulation results are plotted in Figure 4.6. As expected, ALL-NC, equipped with over three times as many network coding enabled wireless access points as 2-Tier, delivers the most packets. For Toulouse topology, 2-Tier delivers almost as many messages as ALL-NC, with only a third of the APs. However, the performance of 2-Tier is reduced for Helsinki and clearly as low as Baseline for Paris.

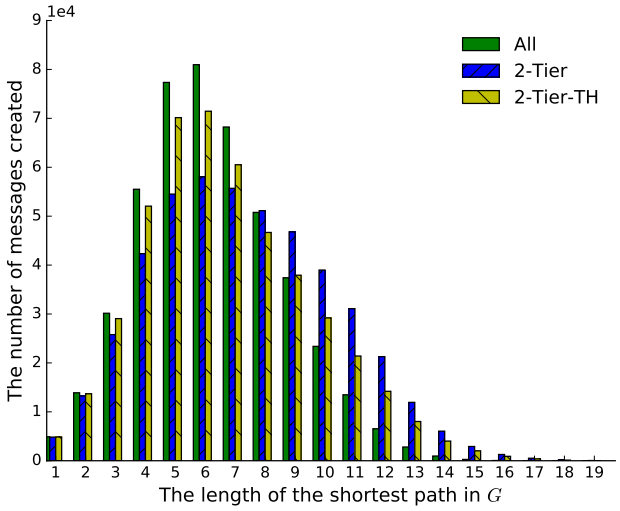
Reasons of performance loss To better understand the cause of this mixed performance, we have plotted the distribution of messages delivered with respect to the path length (in G) in Figure 4.7. Baseline and ALL-NC have the same number of access points so the routes selected are the same and are shown in Figure 4.7 under the label "ALL". The data shows that for Toulouse, on average, the packets were delivered over routes of similar hop count for all architectures, explaining the similar performance of ALL-NC and 2-Tier in terms of packets delivered.

For Paris and Helsinki, however, the 2-Tier's reduction in deployed access points does lead to packets taking longer paths, explaining why the number of packets delivered drops when compared to ALL-NC. Indeed, flows are routed over longer paths and thus take more time to arrive at their destination. Recall that we measure the number of messages that arrive within 12 hours at their destination.

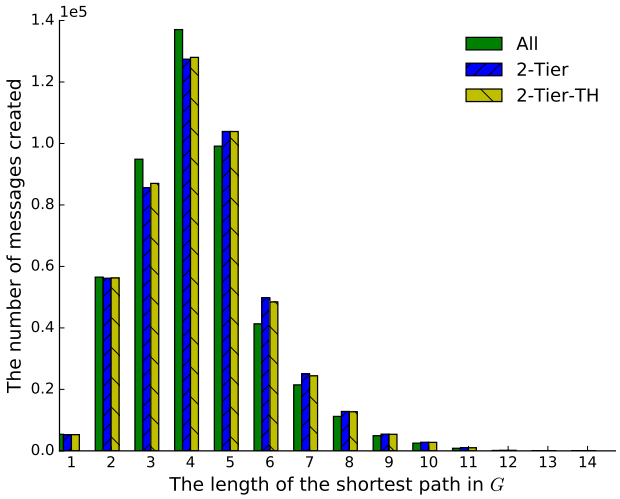
Figure 4.4 can be used to illustrate this natural stretch of path using 2-tier compared to ALL-NC forwarding. In ALL-NC, a message m from a can be directly sent to b. But in 2-Tier, m is delivered to the destination with three more hops following the path $a \to c \to e \to d \to b$ as b doesn't belong to the MCDS. This increase in path length is not visible for Toulouse because forwarding messages



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.7 – The path length in G of the routes taken by the messages created during the simulation.

using its original topology with ALL-NC over shortest paths resumes to using the MCDS.

Station id	Degree	Station id	Degree	Station id	Degree
323	2	326	2	301	2

(a) Toulouse

Station id	Degree	Station id	Degree	Station id	Degree
3234	5	3241	2	3071	2
3189	4	3154	2	3076	2
3124	3	3265	2	3173	2
3219	3	3088	2	3221	2
3132	2	3087	2		

(b) Paris

Station id	Degree	Station id	Degree	Station id	Degree
1724	5	1673	2	1531	2
1543	4	1703	2	1631	2
1702	4	1707	2	1599	2
1633	4	1556	2	1625	2
1626	3	1663	2	1519	2
1610	3	1718	2	1616	2
1600	3	1520	2	1566	2
1604	3	1640	2	1689	2
1688	3	1725	2	1684	2
1547	2	1720	2	1577	2

(c) Helsinki

Table 4.3 – All non-CDS stations with $deg(v) \geq 2$ in PTNs.

Relationship with node degree The pursuit of a *minimum* connected dominating set for a graph G most likely excludes some very important nodes from the CDS for network coding, leading to significant decrease in network performance in our delay tolerant context. Network coding has been shown in our context to be highly beneficial if flows cross two-by-two at a bus station in Chapter 3. This is the case if in a one-hop neighborhood of a node i, one or more pairs of neighbors exchange messages. Intuitively, the higher the degree of a node is, the more likely such crossflows are.

Table 4.3 lists all non-CDS stations whose degree is greater than or equal to 2 for the three cities. As we can see, some stations with high degree are excluded from CDS. Our aim in the next part is to ensure high degree nodes are preserved in the CDS calculation such as to keep the full benefit of network coding. To retain the critical nodes while constructing a CDS, we propose a new CDS calculation heuristic, called **TH-CDS**, that we develop in the next section.

The fact of keeping high degree nodes should be beneficial for two reasons: *i*) the average length of paths will be reduced, and *ii*) the likelihood of crossflows existence be increased, leading to a straightforward network coding benefit.

4.2.4 2-Tier-TH

In this section, we propose a novel CDS calculation heuristic named TH-CDS that ensures high degree nodes are kept in the CDS to avoid the loss of performance presented earlier. The architecture that leverages this new TH-CDS calculation is referred to as 2-Tier-TH in the following.

The main idea of 2-Tier-TH is to construct a CDS and meanwhile retain the nodes whose degree is greater than or equal to a fixed θ integer value. The pseudo-code of TH-CDS is listed in Algorithm 4.3. It includes the following steps:

1. Categorize nodes in $G(V, E)$ into three types: *i*) leaf nodes L with degree 1; *ii*) nodes V' whose degree is greater than or equal to the given threshold value θ, i.e. $deg(v) \geq \theta$; *iii*) the rest of the nodes R, i.e. $R = V - L - V'$.
2. Remove leaf nodes L from V one by one. Whenever a leaf node v is removed, add the neighbor of v with the highest degree to V' if v is not yet dominated.
3. Check the rest of nodes R (i.e. $V - L - V'$). Order nodes of R by ascending degree. For each node v in R, if $G(V - \{v\})$ is disconnected, v must be a part of CDS; otherwise, it can be excluded from CDS if v has been already dominated by one of V' and if not, add the neighbor of v with the maximum degree to V'.
4. In the end, V' forms a connected dominating set.

The connectivity of the subgraph G' induced by V' is guaranteed as *i*) removing leaf nodes from G does not affect the graph connectivity; *ii*) nodes with $deg(v) \geq \theta$

Algorithm 4.3 TH-CDS: CDS with a given threshold

Require: A connected graph $G(V, E)$ and a threshold value θ.
$L \leftarrow \{v|deg(v) = 1\}$, get leaf nodes.
$V' \leftarrow \{v|deg(v) \geq \theta\}$, add nodes with $deg(v) \geq \theta$ to the CDS V'.
$R \leftarrow V - L - V'$, get the rest of the nodes of G.
for all $v, v \in L$ **do** ▷ Remove leaf nodes from V and add their dominating node to V'.
  if $N(v) \cap V' = \varnothing$ **then**
    $V' \leftarrow V' \cup$ the neighbor of v with the highest degree.
  end if
  $V \leftarrow V - \{v\}$, remove v from V.
end for
for all $v, v \in R$ **do** ▷ Check R on ascending order by degree.
  if $v \notin V'$ **then**
    if $G(V - \{v\})$ is connected **then**
      if $N(v) \cap V' = \varnothing$ **then**
        $V' \leftarrow V' \cup$ the neighbor of v with the highest degree.
      end if
      $V \leftarrow V - \{v\}$, remove v from V.
    else
      $V' \leftarrow V' \cup \{v\}$
    end if
  end if
end for

are kept in V' at the beginning; *iii*) a node in R is removed from G only when $G(V-\{v\})$ is connected. Figure 4.8 shows the CDS (highlighted in red) calculated with Algorithm 4.3 on $\theta = 4$ for our PTNs.

City	Nodes (V)	L ($deg(v) = 1$)	V' [1]	R (Others)
Toulouse	44	28	7	9
Paris	213	114	38	61
Helsinki	217	127	27	63

Table 4.4 – Number of nodes in different categories in 3 different cities at the initialization phase.

Unlike the construction method in 2-Tier that always checks the connectivity of the remaining graph whenever a node is removed from G [2], in 2-Tier-TH, we only check the connectivity of the graph whenever a node in R is removed. This is because leaf nodes L can be removed from V directly and high degree nodes are pre-added to V' directly. As such, it leads to a dramatic reduction of computation time, especially for sparse graphs such as our PTNs.

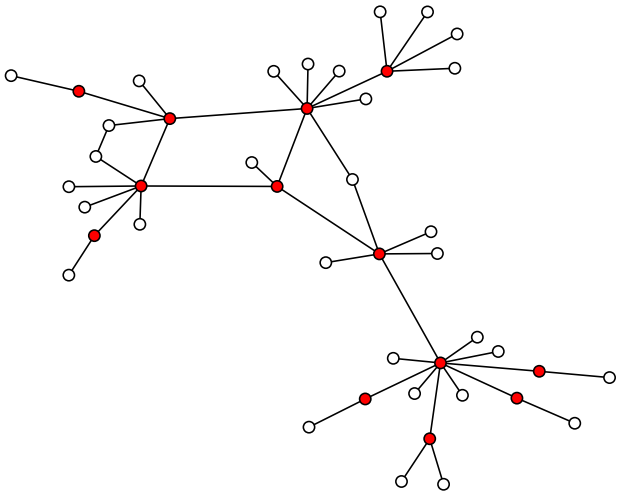
Table 4.4 lists the number of nodes in different categories in three different cities at the initialization phase when $\theta = 4$. Compared to MCDS, TH-CDS reduces the complexity by 79.55%, 71.36% and 70.97% in Toulouse, Paris and Helsinki respectively. Table 4.5 shows the size of MCDS compared to TH-CDS. Even though our TH-CDS heuristic hasn't been designed to purely minimize the number of nodes in the CDS, it still provides a set size that is really close to the MCDS one.

City	MCDS	TH-CDS
Toulouse	13	13
Paris	85	87
Helsinki	60	64

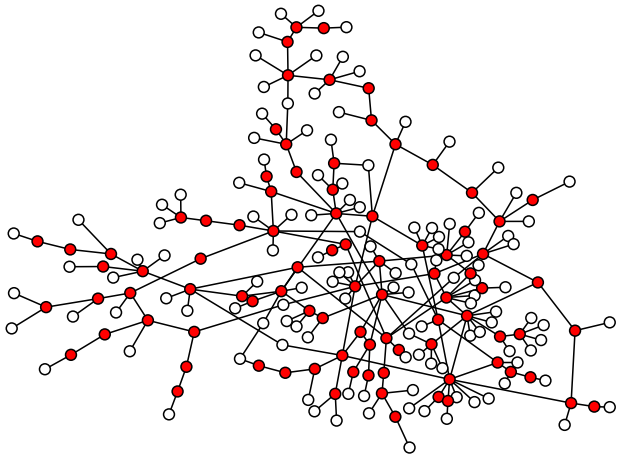
Table 4.5 – Number of nodes that belong to MCDS and to TH-CDS

Similarly to 2-Tier, the routing table should be updated in 2-Tier-TH as well (using Algorithm 4.2). With the new routing table, again, we plot the CDF of

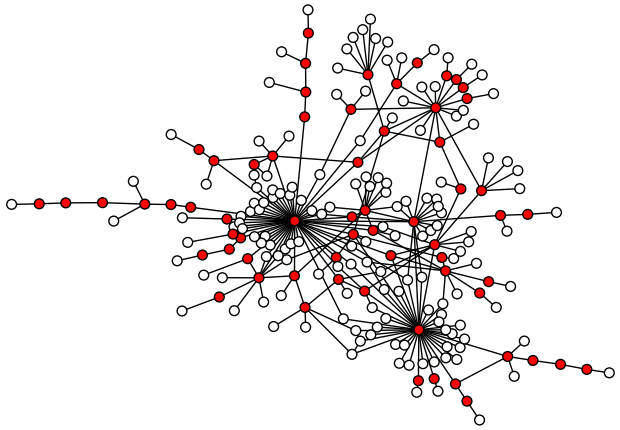
1. At the initialization phase.
2. It takes $O(V + E)$ times to check the connectivity of a graph $G(V, E)$.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.8 – A CDS (in red) calculated with Algorithm 4.3 in three PTNs ($\theta = 4$).

shortest route length in G in Figure 4.7. Data shows that for Toulouse, the distribution of the hop count in 2-Tier-TH is the same as in 2-Tier because both heuristics generate the same CDS. For Paris and Helsinki, however, with critical nodes kept in 2-Tier-TH, messages do take shorter paths than 2-Tier. More importantly, their distribution gets close to that of ALL-NC in particular in the Helsinki topology. A finer performance evaluation of 2-Tier-TH is given in Section 4.4.2.1 and Section 4.4.2.2.

4.3 3-Tier Architecture

Installing network coding enabled APs requires more expensive and dedicated hardware to support advanced processing, such as XOR operation on the fly, secure communication (resistance to so-called pollution attacks). In order to further reduce the overall deployment, we propose herein a 3-Tier architecture by dividing stations into three categories where we deploy different types of APs for them [1] :

1. $V''(V'' \in V')$, network coding enabled wireless AP
2. $V' - V''$, regular wireless AP
3. $V - V'$, no wireless AP

As such, the 3-tier architecture is composed of a first tier equal to V'', a second tier equal to $V' - V''$ and a third tier equal to $V - V'$. Clearly, the key issue is how to select a subset V'' from the CDS nodes V'.

Motivated by the well known 80-20 rule, we basically want to select V'' from V' to offer the same performance as the one of 2-Tier-TH. Thus, the criterion to select the subset is crucial to the network performance. As we have explained in Chapter 3, the large benefits of network coding can be achieved if a station has a lot of crossflows.

Using the same intuition than for the design of TH-CDS, we analyze in this chapter whether selecting highly central nodes in the graph can help us capture, in a simple manner, the nodes that are essential for network coding. In graph theory, several centrality metrics are defined to rank nodes by their importance in

1. V' is calculate with TH-CDS introduced in Section 4.2.4.

the graph regarding different criteria, the node degree being one of them. Next section will introduce the ones we have investigated in this last contribution.

4.3.1 Node centrality

The centrality of a vertex measures its relative importance within a graph. It is essential for many applications, such as identifying the most influential person(s) within a social network, critical infrastructure nodes in the Internet and super-spreaders in a disease spread network.

In a computer network, typically, a central node has a stronger capability of connecting to other network members. It can be effectively used to design routing protocols thus enhance delivery performance, such as SimBetTs [77], Bubble Rap [78]. In this work, we leverage node centrality to decide which bus stations should be equipped with a wireless transceiver that performs network coding so as to reduce the overall deployment cost.

An abundance of metrics has been designed to measure the node centrality based on different concepts, including shortest-path [79], max flows [80], random walks [81] and current flows [82]. In this work, we focus on three widely used metrics in computer networks. When introducing these metrics, we consider a standard $G(V, E)$ graph notation.

4.3.1.1 Degree centrality

The degree centrality for a node v is the number of edges incident to v, i.e. the degree of v, $deg(v)$.

$$C_D(v) = deg(v) \tag{4.1}$$

Figure 4.9-(a) illustrates the degree centrality in a graph. Node f has the lowest degree centrality, which is 1. In our scenarios, a station with higher degree means that more bus lines cross at the station. Naturally, more buses operated on these bus lines should offer more crossflows opportunities.

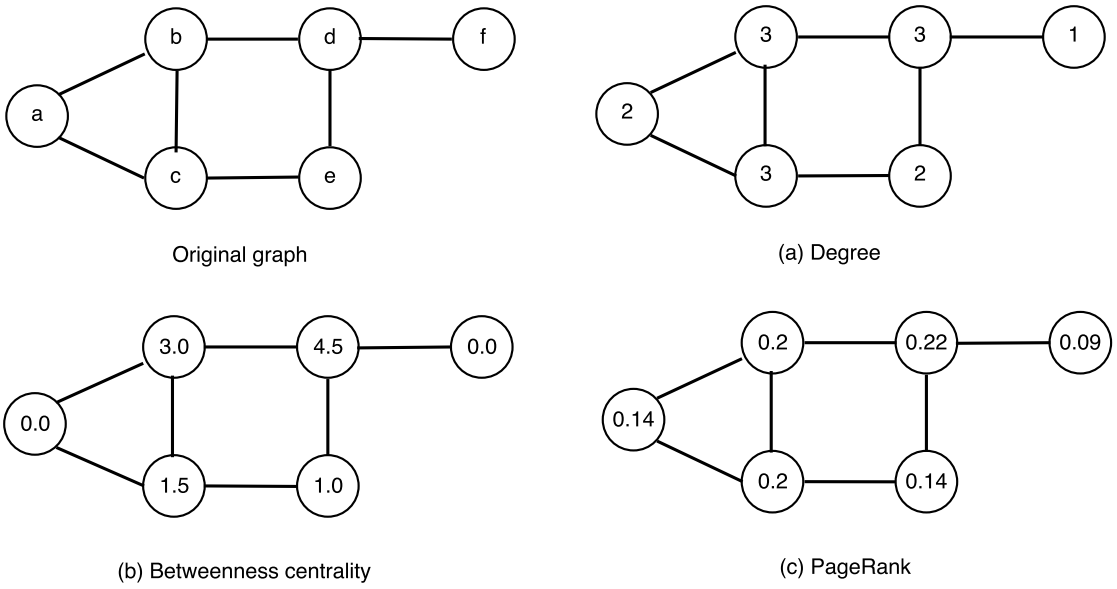


Figure 4.9 – An illustration of node centrality in a graph with different metrics.

4.3.1.2 Betweenness centrality

A natural distance metric between node s and node t is their shortest path which is the minimum of the sum of the weights of edges of all paths between s and t. Numerous routing protocols are based on the shortest path, such as Open Shortest Path First (OSPF).

Betweenness centrality [79] of a node v is the sum of the fraction of all-pairs [1] shortest paths that pass through v, denoted by $C_B(v)$.

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}, s \neq v \neq t \tag{4.2}$$

where $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|v)$ is the number of those paths passing through node v. If $v \in s,t$, $C_B(v) = 0$.

As shown in Figure 4.9-(b), node labels present the betweenness centrality. Node d has the highest betweenness centrality. Table 4.6 shows how to calculate the betweenness of node b. For instance, for the pair of node (c,d), there are two shortest paths between c and d (i.e. $c \rightarrow b \rightarrow d$ and $c \rightarrow e \rightarrow d$) and one of them passes through b. Thus, we have $\frac{\sigma(c,d|b)}{\sigma(c,d)} = 0.5$.

1. All the combinations (not permutations) for undirected graphs.

(s,t)	All shortest paths	$\sigma(s,t\|b)/\sigma(s,t)$
(a,d)	$a \to b \to d$	1
(a,f)	$a \to b \to d \to f$	1
(c,d)	$c \to b \to d$ $c \to e \to d$	0.5
(c,f)	$c \to b \to d \to f$ $c \to e \to d \to f$	0.5
In total:		3.0

Table 4.6 – Calculation of the betweenness centrality of node b in Figure 4.9.

In computer networks, betweenness centrality can be regarded as how much a node can facilitate communication to other nodes. A node with higher betweenness centrality has more capability of facilitating communication. It identifies as well a bottleneck node.

In our scenarios, routing tables for each station are calculated based on the shortest path. A station with higher betweenness centrality means more data flows going through that station. We have shown in Chapter 3 that network coding can bring large benefits for the stations that experience lots of crossflows. Thus, it seems reasonable to select the stations with high betweenness centrality to install network coding enabled APs as well.

4.3.1.3 PageRank

PageRank [83] was originally designed by Google founders to rank web pages in their search engine results. The World Wide Web can be modeled as a directed graph $G(V,E)$ where edges represent hyperlinks and vertices represent corresponding web pages. Formally, there is a directed edge (p_i,p_j) in E if there exists page p_i linking to page p_j. PageRank assigns a numerical weighting to each web page p_i in V, which is referred to as the PageRank of p_i and denoted by $PR(p_i)$, indicating how important a page is.

The PageRank of a page is determined by the number and quality of links to the page. We demonstrate the underlying core ideas of PageRank by the case of Figure 4.10. *i*) The number of links to a page. Page B receives the largest number of links and its PageRank is quite high, reaching at 38.4%. Using citation networks

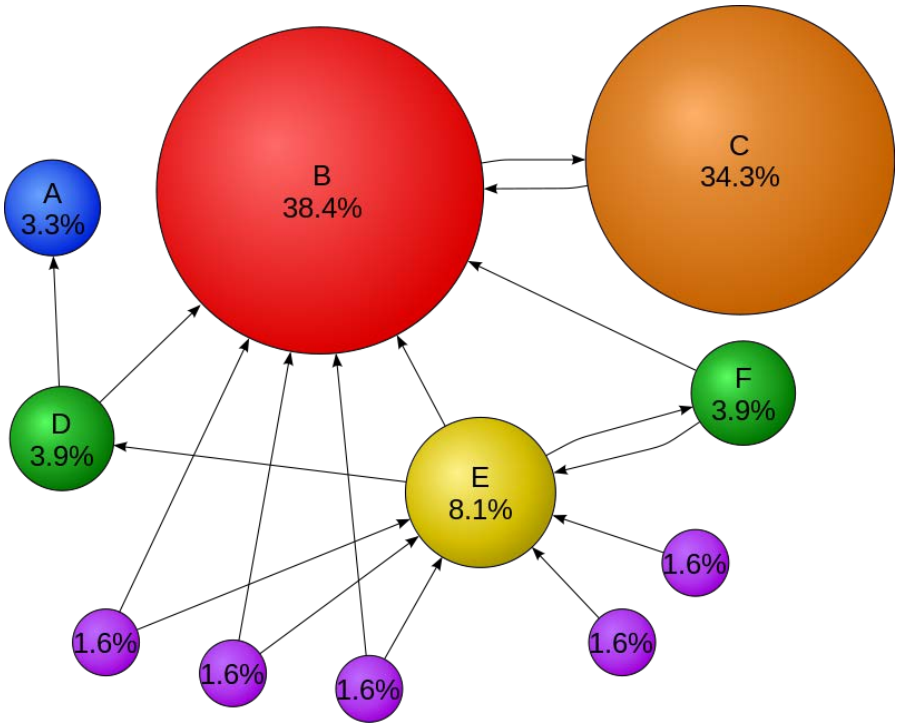


Figure 4.10 – PageRanks for a simple network (image from Wikipedia).

as an analogy, the more an academic paper is cited, the more important the paper is supposed to be. *ii*) The quality of these links. Page C has a higher PageRank than Page E, even though there are fewer links to C. This is because the one link to C comes from an important page (i.e. B) and hence is of high value. Using citation networks as an analogy, a paper cited by an important paper is supposed to be a great paper.

When calculating PageRank, Page *et al.* consider a random web surfer who starts on a random page (say p_i) and proceeds to a randomly chosen web page that p_i hyperlinks to. The PageRank score of p_i is divided evenly among all outbound links. For instance, we assume the surfer is currently at page E in Figure 4.10, out of which there are three hyperlinks to pages D, B, F. At the next time step, the surfer clicks one of these three pages with equal probabilities, i.e. 1/3. The PageRank theory holds that the surfer will eventually stop clicking. The probability that the surfer will continue is given by a damping factor d[1]. The

1. The value recommended in the original paper [83] is 0.85.

PageRank of p_i is the product of the damping factor and the sum of the incoming PageRank scores, and then added to $(1-d)/N$ [1] where $N = |V|$, the number of vertices in G. That is,

$$\begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix} \tag{4.3}$$

Eq. (4.3) can be expressed as a matrix equation.

$$\mathbf{R} = \frac{1-d}{N}\mathbf{1} + d \cdot \mathbf{M} \cdot \mathbf{R} \tag{4.4}$$

where $\mathbf{1}$ is the column vector of length N containing only ones and,

$$\mathbf{M}_{ij} = \ell(p_i,p_j) = \begin{cases} 1/L(p_j), & \text{if } j \text{ links to } i. \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

$L(p_j)$ denotes the number of outbound links on page p_j. Actually, $\mathbf{M}$ can be calculated by $\mathbf{M} = (\mathbf{K}^{-1}\mathbf{A})^T$ where $\mathbf{A}$ denotes the adjacency matrix of G and $\mathbf{K}$ is the diagonal matrix with the outdegrees in the diagonal.

The solution of Eq. (4.4) exists and is unique for $0 < d < 1$ [83]. The solution is given by Eq. (4.6) with the identity matrix $\mathbf{I}$.

$$\mathbf{R} = (\mathbf{I} - d\mathbf{M})^{-1}\frac{1-d}{N}\mathbf{1} \tag{4.6}$$

The result obtained from Eq. (4.6) is normalized, i.e. $\mathbf{R}^\star = \mathbf{R}/|\mathbf{R}|$. As an example, Figure 4.10 shows PageRanks expressed as percentages. The size of each node is proportional to the total size of the other nodes which are pointing to it.

In our scenarios, a station that serves more bus lines is likely to relay more messages and benefits from network coding. These bus lines can be seen as incom-

1. Pages with no outbound links are assumed to link out to all other pages in the collection.

Station id	Degree centrality	In CDS [1]
337	10	Y
298	8	Y
327	7	Y
340	6	Y
311	5	Y
304	5	Y
307	4	Y
330	3	Y
323	2	N
321	2	Y
326	2	N
312	2	Y
328	2	Y
336	2	Y
301	2	N
332	2	Y
...	...	...

Table 4.7 – The degree centrality of stations in the Toulouse topology.

ing links to the station. As pinpointed earlier, the more incoming links to a node, the higher PageRank the node has. PageRank is a way of measuring the relative importance of nodes within a directed graph. In this work, we leverage PageRank to measure the importance of stations and select some of the most important stations to equip with network coding enabled APs. Since PageRank is applied to directed graphs, our undirected graph is turned into a directed graph by assigning two directions to each edge. As an example, node labels in Figure 4.9-(c) present the value of PageRank.

4.3.2 3-Tier selection

We calculate node centrality defined in Section 4.3.1 for all nodes in a graph $G(V, E)$ and select the n nodes V'' ($V'' \in V'$) with the highest values. Take the Toulouse topology as an example, the degree centrality is calculated for all stations in the whole graph. The results are listed in Table 4.7 on descending order of degree

1. It is calculated using 2-Tier-TH

centrality. We select the top n nodes that belong to CDS as V''. For instance, V'' equals $\{337, 298\}$ when $n = 2$.

On the example of Figure 4.9, the three centrality metrics select the same top node, but the ranking of the other ones differ. It is not clear to decide which metric would suit the best to identify the best network coding nodes. Section 4.4.3 analyses by simulation their impact for the three cities of Toulouse, Helsinki and Paris. Similarly, it investigates the impact of n.

4.4 Performance evaluation

In this section, we evaluate the performance of the proposed architectures in terms of the message delivery and cost effectiveness. In summary, we make the following main observations:

1. In Section 4.4.2, we show that 2-Tier-TH not only avoids the loss of performance presented in 2-Tier but also improves the cost effectiveness up to 28%. More importantly, 2-Tier-TH is able to reduce the number of wireless access points required to cover three major cities by approximately a factor of 3 while its performance gets close to ALL-NC.
2. In Section 4.4.3, we conclude that 3-Tier achieves the highest cost effectiveness using degree centrality. We also show that 3-Tier reduces the number of wireless access points capable of performing network coding required to cover three major cities by over an order of magnitude. It accomplishes this cutback while delivering 30% more messages than Baseline.
3. In Section 4.4.4, we show that 3-Tier improves the cost-effectiveness by 71% on average when compared to the closest competitor.
4. In Section 4.4.5, we show that our cost-effective infrastructure can offload a large amount of mobile data. This is especially the case for the Paris topology where up to 4.7 terabytes can be offloaded within 12 hours.

4.4.1 Simulation setup

Simulations are carried out using the same simulation setup described in Section 2.5.1. For all three topologies, routing tables are generated before the simula-

tion starts so that messages are relayed based on predefined routes. We compare the performance among 5 proposed architectures which are:

- **Baseline** introduced in Chapter 2 where all stations are equipped with a regular wireless AP and forward packets without performing network coding.
- **ALL-NC** introduced in Chapter 3 where all stations are equipped with a network coding enabled wireless AP.
- **2-Tier** introduced in Section 4.2.2 where only CDS stations (selected with MCDS heuristic) are equipped with a network coding enabled wireless AP.
- **2-Tier-TH** introduced in Section 4.2.4 where only CDS stations (selected with TH-CDS heuristic) are equipped with a network coding enabled wireless AP.
- **3-Tier** introduced in Section 4.3 where a subset V'' of a connected dominating set V' are equipped with a network coding enabled wireless AP and the stations of $\{V' - V''\}$ are equipped with a regular wireless AP. V' is composed of CDS stations selected with TH-CDS heuristic. Elements of V'' are selected following different criteria in 4.4.3.

4.4.2 Evaluation of 2-Tier-TH

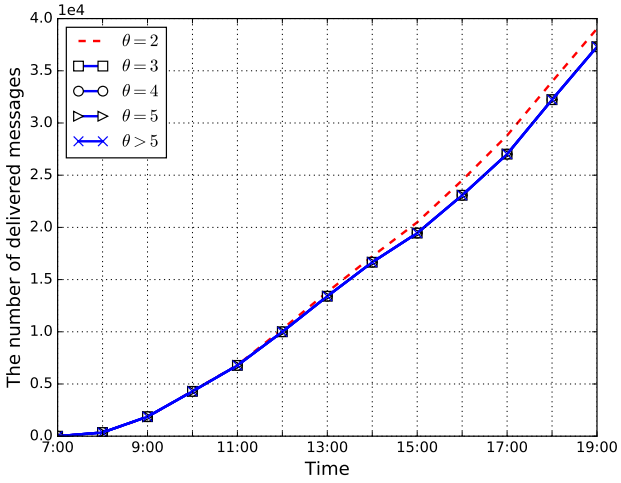
In this section, we investigate first the impact of θ in the TH-CDS calculation to find out which one achieves the highest cost effectiveness. Next, we compare the performance of 2-Tier-TH to 2-Tier, Baseline and ALL-NC architectures.

4.4.2.1 Evaluation the effect of θ in 2-Tier-TH

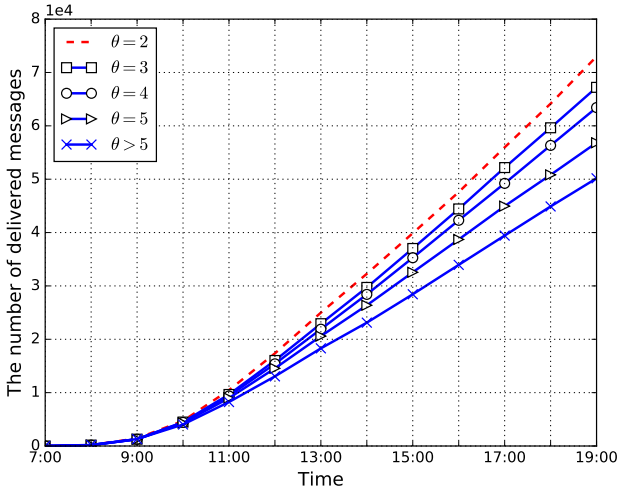
In this subsection, we evaluate the effect of the threshold value θ in 2-Tier-TH in terms of the message delivery and the cost effectiveness.

Method: We calculate the CDS for our PTNs using 2-Tier-TH with all possible threshold values. Different CDS calculations correspond to different routing tables. To find the best threshold value for 2-Tier-TH, we define the cost effectiveness as the ratio between the number of messages delivered to the size of the corresponding CDS, i.e. the number of delivered messages per CDS node.

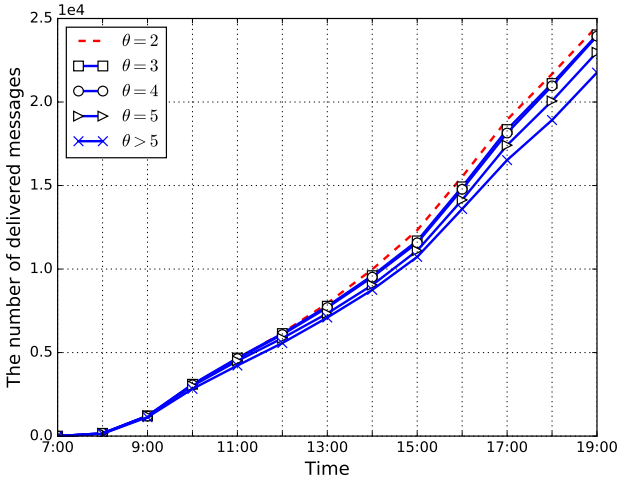
Results: Table 4.8 shows the number of wireless APs to cover three cities for



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.11 – Number of messages delivered for Baseline, ALL-NC, 2-Tier-TH with different θ.

City	2-Tier-TH				
	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 5$	$\theta > 5$
Toulouse	16	13	13	13	13
Paris	99	89	87	86	85
Helsinki	90	69	64	61	60

Table 4.8 – Number of wireless access points required to cover 3 different cities for 2-Tier-TH with different θ.

2-Tier-TH with different threshold values θ. It is easy to understand that the size of the CDS decreases with the increase of θ as the higher θ, the fewer nodes are pre-added to CDS. As explained in Section 4.2.4, messages take shorter paths with more nodes in CDS, leading to the increase of messages delivered, as demonstrated in Figure 4.11.

The threshold value θ begins with 2 as leaf nodes do not have network coding probability [1]. In the Toulouse topology, the connected dominating sets are the same when $\theta > 2$. This is because all nodes with $deg(v) > 2$ are already included in a minimum CDS, as shown in Table 4.3-(a). For the same reason, The connected dominating sets are the same when $\theta > 5$ in Paris and Helsinki, as shown in Table 4.3-(b) and -(c) respectively. Actually, 2-Tier-TH degenerates to 2-Tier when $\theta > 2$ in Toulouse and $\theta > 5$ in Paris and Helsinki. When $\theta = 2$, 2-Tier-TH has the same performance as ALL-NC as network coding can not bring any benefit to leaf stations ($deg(v) = 1$).

Figure 4.12 shows the cost effectiveness of 2-Tier-TH with different threshold values. The connected dominating sets are the same in Toulouse when $\theta > 2$, explaining that they have the same cost effectiveness. For the topologies Paris and Helsinki, the cost effective of 2-Tier-TH is higher than that of 2-Tier (i.e. 2-Tier-TH with $\theta > 5$), with an improvement of 28% (when $\theta = 3$) and 4% (when $\theta = 4$) respectively.

4.4.2.2 Evaluation of 2-Tier-TH

We learnt from Section 4.4.2.1 that 2-Tier-TH achieves the highest cost effectiveness at $\theta > 2$, $\theta = 3$ and $\theta = 4$ in Toulouse, Paris and Helsinki respectively. In

1. When $\theta = 1$, 2-Tier-TH degenerates to ALL-NC.

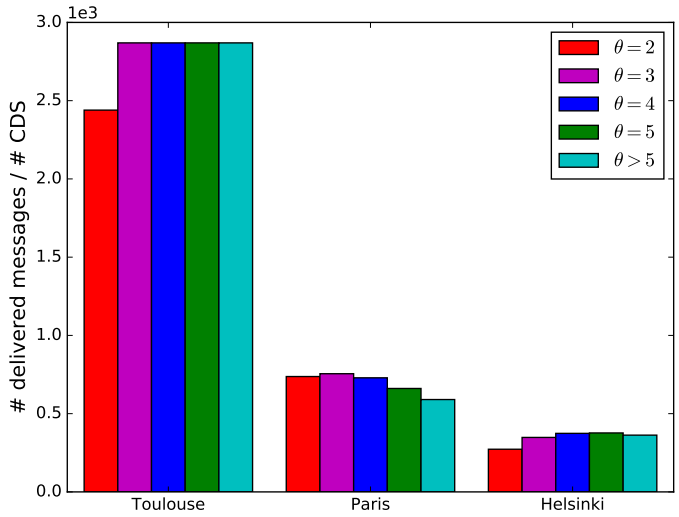


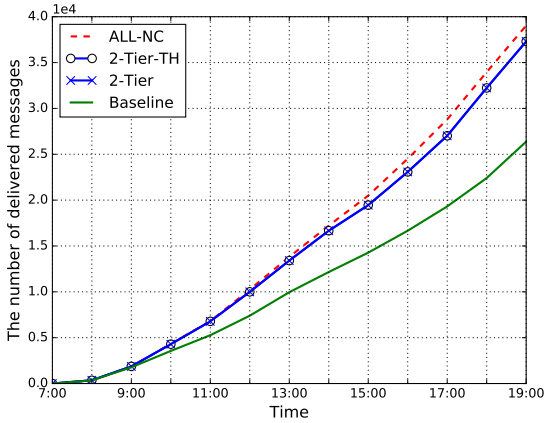
Figure 4.12 – The number of delivered messages per node in 2-Tier-TH with different θ.

this subsection, we compare the performance of 2-Tier-TH with 2-Tier, Baseline and ALL-NC in terms of the cost of its deployment and the packet delivery.

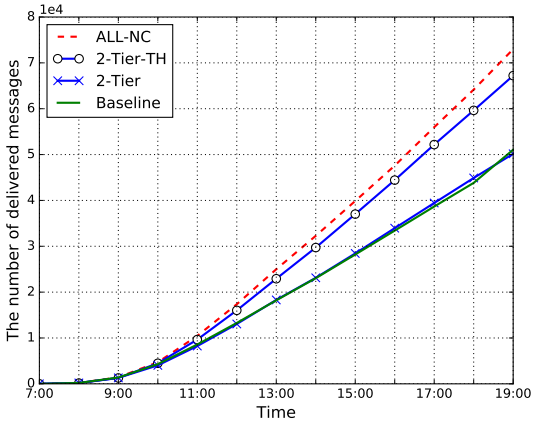
City	Baseline	ALL-NC	2-Tier	2-Tier-TH
Toulouse	44	44	13	13 ($\theta > 2$)
Paris	213	213	85	89 ($\theta = 3$)
Helsinki	217	217	60	64 ($\theta = 4$)

Table 4.9 – Number of wireless access points required to cover 3 different cities.

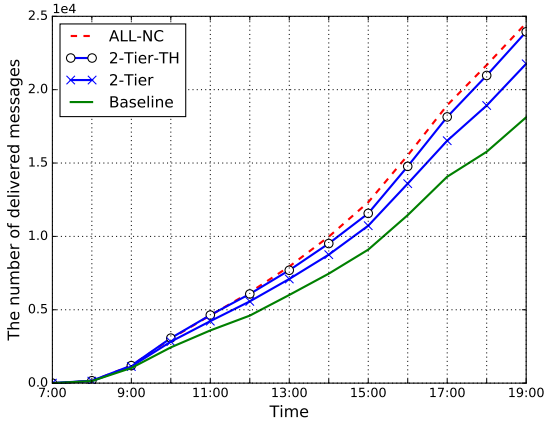
Results: Table 4.9 shows the number of wireless access points necessary to cover three major cities. The results show that both 2-Tier and 2-Tier-TH reduce the required number of APs by approximately a factor of 3, compared to Baseline and ALL-NC. These results are more impressive when looking into the number of messages delivered, shown in Figure 4.13. The performance of 2-Tier-TH gets close to that of ALL-NC. Compared with 2-Tier, 2-Tier-TH increases the number of delivered messages by 34% and 10% in Paris and Helsinki respectively by adding just a few more APs (4 APs more in Paris and Helsinki respectively). These significant improvements are because 2-Tier-TH makes sure important nodes remain in the CDS, as explained in Section 4.2.4. As such, messages take shorter paths than 2-Tier, explaining why the number of delivered messages increases.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.13 – Number of messages delivered for Baseline, ALL-NC, 2-Tier-TH and 2-Tier.

4.4.3 Evaluation of 3-Tier

In this section, we evaluate first the impact of n in 3-Tier which corresponds to the number of nodes of V'' where network coding enabled access points are to be deployed. Next, we look at the impact of the centrality measure chosen to select the n nodes to deploy network coding at.

4.4.3.1 Evaluation of the effect of n in 3-Tier

We learnt from Section 4.4.2.1 that 2-Tier-TH achieves the highest cost effectiveness at $\theta > 2$, $\theta = 3$ and $\theta = 4$ in Toulouse, Paris and Helsinki respectively. In this subsection, we evaluate the effect of n (the top n nodes with the highest degree, betweenness and PageRank from 2-Tier-TH performing network coding) in 3-Tier in terms of the potential cost. The top n nodes are selected from 2-Tier-TH calculated with $\theta > 2$, $\theta = 3$ and $\theta = 4$ in Toulouse, Paris and Helsinki respectively.

Method: We define the cost effectiveness of an architecture as the ratio between delivered messages and the deployment cost. To quantify the deployment cost without resorting to using specific dollar amounts, we use a simple cost function which assigns the cost of 1 to a simple wireless access point and 2 to a wireless access point capable of performing network coding.

For Toulouse, we evaluate the cost effectiveness on all possible choices of n (i.e. $1, 2, \ldots, 13$). For Paris and Helsinki, we only evaluate on the choices of $1, 2, \ldots, n$ where $n = \lceil 20\% \cdot |V'| \rceil$, i.e. the top 20% nodes with the highest value of the CDS.

Results: Table 4.10 lists the nodes identifiers[1] in descending order of centrality, where centrality is measured by the three considered metrics. Figure 4.14 shows the cost effectiveness for 3-Tier with different n for different metrics. For the simple Toulouse topology, which is close to a star topology, degree and PageRank performance overlap since they select the same set of nodes for V''. For Helsinki and $n \geq 4$, all three metrics select a similar set of nodes for V''. In contrast, different sets are selected by the three metrics for Paris. In this case, there are two solutions, one with 13 nodes selected with Degree and one with 16 nodes selected

1. The station id is normalized for the simulation purpose.

Degree	Betweenness	PageRank
337	340	337
298	337	298
327	298	327
340	307	340
311	327	304
304	311	311
307	304	307
330	330	330
321	321	312
312	312	336
328	328	332
336	336	321
332	332	328

(a) Toulouse

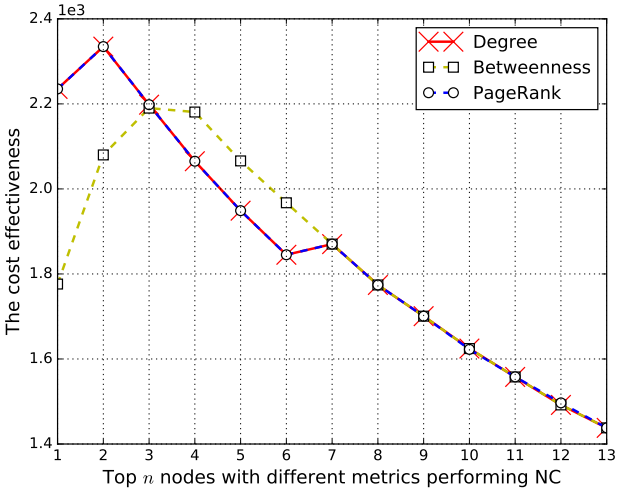
Degree	Betweenness	PageRank
3059	3252	3059
3164	3164	3164
3252	3195	3214
3214	3214	3252
3195	3059	3195
3227	3234	3056
3056	3056	3227
3075	3131	3179
3131	3172	3122
3240	3169	3075
3122	3230	3105
3123	3147	3240
3147	3189	3117
3169	3075	3147
3086	3227	3086
3105	3123	3165
3179	3127	3131
3157	3179	3230

(b) Paris

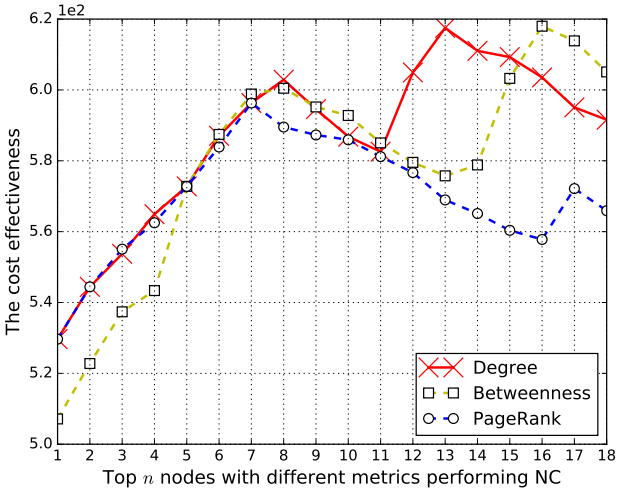
Degree	Betweenness	PageRank
1658	1658	1658
1512	1512	1512
1544	1544	1680
1680	1680	1544
1553	1559	1553
1674	1541	1537
1568	1568	1674
1593	1642	1568
1537	1537	1593
1704	1633	1704
1559	1618	1629
1541	1553	1559
1555	1543	1541

(c) Helsinki

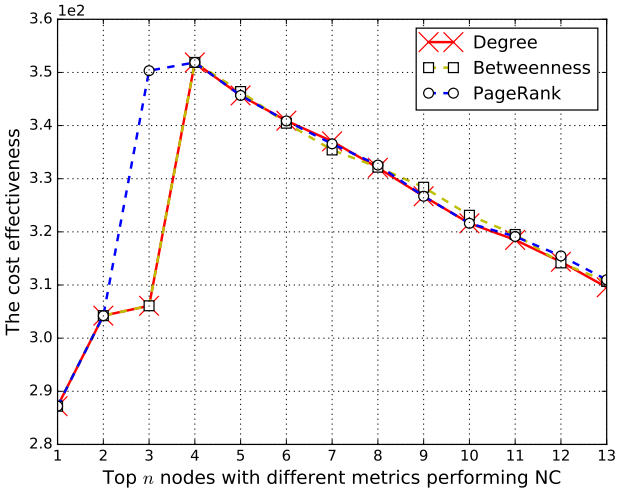
Table 4.10 – Top n nodes with different metrics in three cities.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.14 – The cost effectiveness of 3-Tier with the top n nodes in different metrics performing NC. In Toulouse and Helsinki, Degree and PageRank performance clearly overlap.

with Betweenness that perform equally. Both solutions could be further investigated in future works to better understand the impact of the metrics. Compared to Toulouse and Helsinki, the Paris network is closer to a mesh, providing a larger set of solutions that really offer different routes in the network.

However, we can conclude for these three topologies that we can achieve the highest cost effectiveness with 3-Tier using degree centrality, captured at $n = 2$, $n = 13$ and $n = 4$ in Toulouse, Paris and Helsinki, respectively.

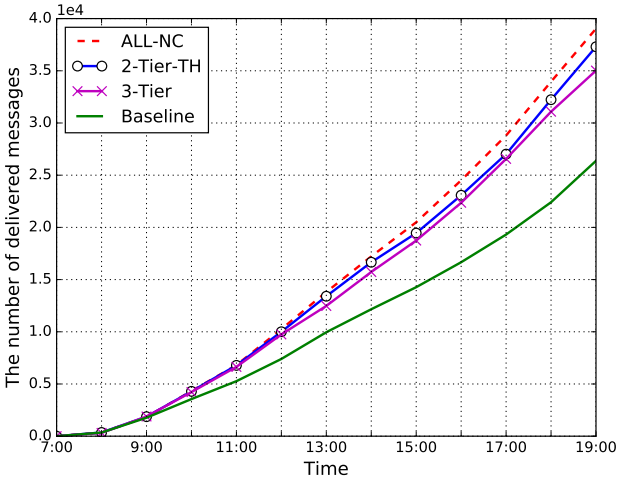
4.4.3.2 Evaluation of 3-Tier

We learnt from Section 4.4.3.1 that 3-Tier achieves the highest cost effectiveness at $n = 2$, $n = 13$ and $n = 4$ in Toulouse, Paris and Helsinki respectively. In this subsection, we evaluate the performance of the 3-Tier architecture in terms of the number of messages delivered and the number of access points deployed.

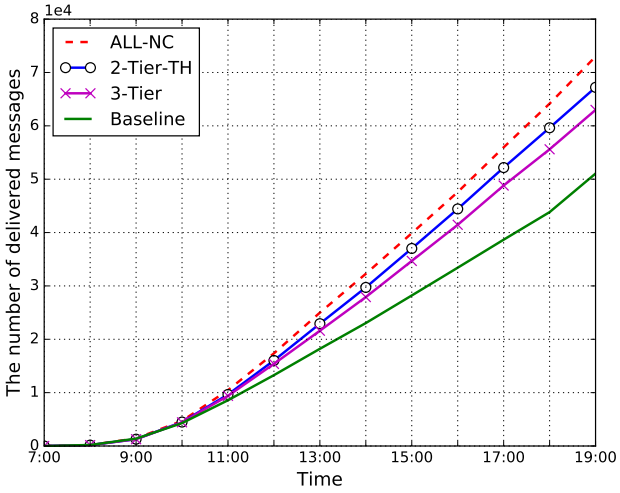
City	2-Tier-TH	3-Tier
Toulouse	13 ($\theta > 2$)	2
Paris	89 ($\theta = 3$)	13
Helsinki	64 ($\theta = 4$)	4

Table 4.11 – Number of wireless access points capable of performing network coding required to cover 3 different cities. The 3-Tier architecture reduces the number of such interfaces by over an order of magnitude.

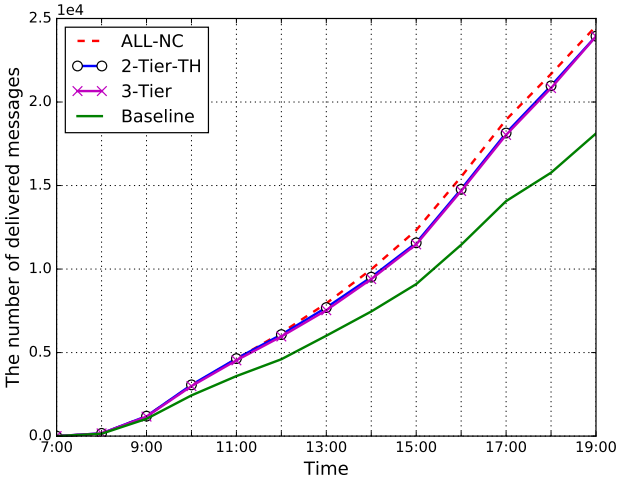
Results: Table 4.11 shows the number of wireless access points capable of performing network coding for 2-Tier-TH and 3-Tier. Data shows that 3-Tier reduces the number of access points by over an order of magnitude, which as we show in Section 4.4.4 has the potential to dramatically reduce the cost of deployment. Fortunately, this significant cutback in infrastructure does not affect performance. Figure 4.15 shows that 3-Tier delivers essentially the same number of messages as 2-Tier-TH. Compared to Baseline, 3-Tier increases the number of delivered messages by 32.79%, 23.36% and 32.07% in Toulouse, Paris and Helsinki respectively.



(a) Toulouse



(b) Paris



(c) Helsinki

Figure 4.15 – Packets delivered for ALL-NC, 2-Tier-TH, 3-Tier and Baseline.

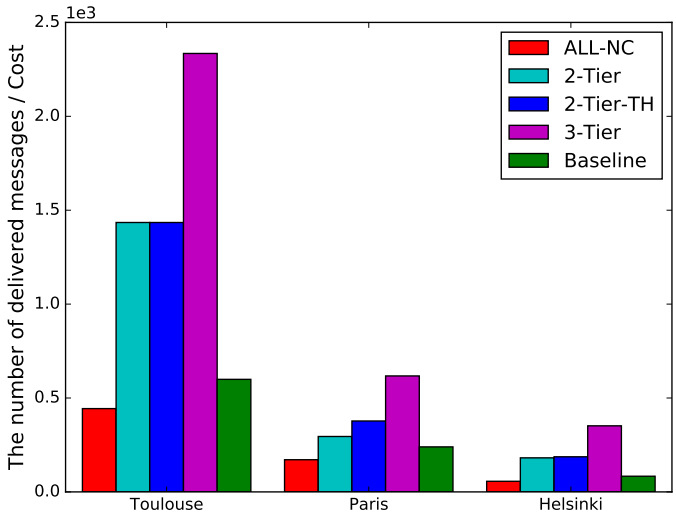


Figure 4.16 – The cost effectiveness for all architectures.

4.4.4 Cost-effectiveness analysis

In this section, we evaluate the potential cost of all proposed architectures for PTNs. The definition of the cost effectiveness is given in Section 4.4.3.1.

Results: Figure 4.16 shows the cost effectiveness for all architectures. Clearly, 3-Tier has the highest cost effectiveness among them. 3-Tier improves the cost effectiveness by 62.74%, 63.53% and 88.13% for Toulouse, Paris and Helsinki, respectively over its closest competitor.

4.4.5 The amount of data offloaded by 3-Tier

We are interested in how much data can be offloaded using our content delivery infrastructure. As stated in Section 2.5.1, in our simulation setup, one message is transmitted in a second among contending nodes. With a given transmission rate of wireless APs R, we are able to estimate the maximum amount of data D offloaded by our infrastructure with Eq. (4.7) (given in terabytes: TB). Indeed, one message could be transmitted for up to one second at a rate of R (this is a coarse upper bound of course).

$$D = \frac{N \cdot R}{8 \cdot 1000 \cdot 1000} \tag{4.7}$$

where N denotes the number of delivered messages at the end of the simulation.

With the number of messages delivered by 3-Tier given in Section 4.4.3 (i.e. 35021, 62986, 23928 in Toulouse, Paris and Helsinki respectively) and for $R = 600$ $Mbps$ (i.e. the maximum rate of IEEE 802.11n), the upper 3-tier offloading bound is of 2.6, 4.7 and 1.8 TB within 12 hours in Toulouse, Paris and Helsinki respectively.

4.5 Conclusion

We have proposed a cost-effective design to address the challenges involved in implementing our content delivery infrastructure. Our 3-Tier architecture is guaranteed to provide end-to-end connectivity and high packet delivery in a cost-efficient manner. We have evaluated our design choices and the proposed architecture using real traces from the public transportation networks of three major European cities. The results show that the 3-Tier architecture achieves a factor of 3 reduction in the number of APs while delivering 30% more messages than a baseline architecture. It can offload a large amount of mobile data as for instance 4.7 terabytes within 12 hours in the Paris topology.

5 Conclusion and Perspectives

5.1 Conclusion

The goal of this work is to provide a cost-effective content delivery infrastructure to relieve the looming bandwidth crunch in urban areas. Our solution proposes to install WiFi access points on both public bus stations and buses, using the latter as data mules. As such, we create a delay tolerant network capable of carrying content users can access while using the public transportation. The main advantages of our proposed infrastructure include low cost, low message loss and the capability to transfer very large amounts of data traffic. Throughout this thesis, we gradually improve the network performance of our content delivery infrastructure in terms of message delivery and cost effectiveness. Real traces in GTFS from the three European cities of Paris, Helsinki and Toulouse are used to validate our design. Thanks to the popularity of GTFS, our scheme is easy to adapt to other cities.

In Chapter 2, we analyzed the publicly available traces of PTN providers and proposed to deploy wireless APs at end-of-the-line (final) bus stops because of long contact duration with buses. Leveraging the stable underlying topology and predictable PTN schedule, we significantly simplified routing design and, by extension, its overhead. We showed that this leads to a dramatic decrease in message replications and transfers. Trace-based simulations showed that our scheme can improve the number of delivered messages by 4 to 8 times while also reducing the overhead ratio when compared to Epidemic routing.

In Chapter 3, we leveraged XOR network coding to address congestion bottlenecks arising at major transportation hubs. The theoretical analysis on a relay

node indicates that both the delivery probability and the overhead ratio could be improved by up to 50%. Our simulation results corroborated these points, showing, in addition, that the average delay is reduced as well. Furthermore, such a scheme is deployed on our content infrastructure, showing a 35% - 48% improvement in delivered messages through trace-based simulations.

In Chapter 4, we proposed a 3-Tier architecture that is guaranteed to provide end-to-end connectivity, high message delivery and a reduced hardware cost. Simulation results showed that the 3-Tier architecture achieves a factor of 3 reduction in the number of access points while delivering 30% more packets than a baseline architecture. It can offload a large amount of mobile data, for instance, up to 4.7 terabytes within 12 hours in the Paris topology.

5.2 Perspectives

While the viability of our design has been demonstrated through both theoretical analysis and trace-based simulations, there are still opportunities for further enhancements.

First, this thesis simplifies the network traffic model by assuming that each node of the network pushes a constant flow of messages to be routed into the PTN. Messages are generated periodically at every node with a certain period of time and its destination is randomly chosen from possible ones every time. It would be important to further investigate how content is actually updated, requested and fetched. With such analysis, it would be possible to dimension properly the amount of storage required at each station and on each bus.

Second, we model our content delivery infrastructure as a non-weighted graph where an edge exists if there exists a bus service between two end stations. However, edges may not be equally important when considering, for example, bus frequencies on routes and the time buses spend at each station. Furthermore, the potential of public transportation schedules can be further explored to identify the node importance, design more effective route paths, etc.

A The throughput improvements by NC

Table A.1 – The throughput improvements achieved by network coding in Toulouse

line1	line2	t_1	t_{12}	t_2	$r(\%)$	D(GB)	$D_{nc}(GB)$	$G_t(\%)$	Station
10	2	0:22:00	9:46:00	2:00:00	80.49	163.0	309.5	89.88	Cours Dillon
10	109	1:19:00	8:55:00	1:40:00	74.93	193.0	326.75	69.3	Malepère
204	205	2:01:00	2:57:00	1:11:00	47.97	97.5	141.75	45.38	Ayguevives Collège
35	83	0:42:00	9:21:00	1:38:00	80.03	171.75	312.0	81.66	Balma-Gramont
35	20	0:34:00	9:29:00	1:28:00	82.34	167.75	310.0	84.8	Balma-Gramont
35	51	3:09:00	6:54:00	0:44:00	63.99	136.5	240.0	75.82	Balma-Gramont
35	77	3:34:00	6:29:00	1:24:00	56.62	160.25	257.5	60.69	Balma-Gramont
35	A	0:00:00	10:03:00	2:17:00	81.49	150.75	301.5	100.0	Balma-Gramont
2	81	2:23:00	9:15:00	0:20:00	77.3	153.75	292.5	90.24	Université Paul Sabatier
2	78	0:49:00	10:49:00	0:36:00	88.42	189.25	351.5	85.73	Université Paul Sabatier
2	56	2:07:00	9:31:00	0:22:00	79.31	159.25	302.0	89.64	Université Paul Sabatier
2	115	2:09:00	9:29:00	0:16:00	79.69	154.25	296.5	92.22	Université Paul Sabatier
2	34	2:04:00	9:34:00	0:19:00	80.06	157.75	301.25	90.97	Université Paul Sabatier
2	82	9:45:00	1:53:00	0:02:00	16.14	29.75	58.0	94.96	Université Paul Sabatier
50	11	1:24:00	9:41:00	0:38:00	82.65	173.75	319.0	83.6	Basso Cambo
50	49	0:57:00	10:08:00	0:31:00	87.36	175.25	327.25	86.73	Basso Cambo

A. THE THROUGHPUT IMPROVEMENTS BY NC

line1	line2	t_1	t_{12}	t_2	$r(\%)$	D(GB)	$D_{nc}(GB)$	$G_t(\%)$	Station
50	53	1:22:00	9:43:00	0:51:00	81.42	184.0	329.75	79.21	Basso Cambo
50	14	0:28:00	10:37:00	1:14:00	86.2	180.25	339.5	88.35	Basso Cambo
50	48	6:23:00	4:42:00	0:23:00	40.99	87.75	158.25	80.34	Basso Cambo
50	A	0:00:00	11:05:00	1:15:00	89.86	166.25	332.5	100.0	Basso Cambo
50	57	2:16:00	8:49:00	0:56:00	73.37	174.25	306.5	75.9	Basso Cambo
50	58	1:59:00	9:06:00	0:31:00	78.45	159.75	296.25	85.45	Basso Cambo
50	8	1:53:00	9:12:00	0:56:00	76.56	180.0	318.0	76.67	Basso Cambo
81	109	0:36:00	3:09:00	6:55:00	29.53	74.25	121.5	63.64	Castanet-Tolosan
81	78	0:25:00	9:10:00	2:15:00	77.46	156.25	293.75	88.0	Université Paul Sabatier
81	56	2:14:00	7:21:00	2:32:00	60.66	210.75	321.0	52.31	Université Paul Sabatier
81	115	1:32:00	8:03:00	1:42:00	71.34	189.75	310.5	63.64	Université Paul Sabatier
81	34	1:19:00	8:16:00	1:37:00	73.81	183.25	307.25	67.67	Université Paul Sabatier
81	202	0:35:00	3:10:00	7:36:00	27.9	73.75	121.25	64.41	Castanet-Tolosan
81	62	0:20:00	3:25:00	7:58:00	29.16	66.25	117.5	77.36	Castanet-Tolosan
81	82	8:19:00	1:16:00	0:39:00	12.38	48.25	67.25	39.38	Université Paul Sabatier
81	205	0:37:00	3:08:00	7:30:00	27.85	74.75	121.75	62.88	Castanet-Tolosan
83	201	1:06:00	9:52:00	0:40:00	84.81	178.0	326.0	83.15	Saint Orens Lycée
83	79	0:45:00	10:13:00	1:08:00	84.44	187.0	340.25	81.95	Saint Orens Lycée
83	78	0:02:00	10:56:00	1:11:00	89.99	165.5	329.5	99.09	Saint Orens Lycée
83	20	0:47:00	10:12:00	0:45:00	86.93	186.75	339.75	81.93	Balma-Gramont
83	51	3:38:00	7:21:00	0:17:00	65.24	123.0	233.25	89.63	Balma-Gramont
83	77	3:42:00	7:17:00	0:36:00	62.88	136.25	245.5	80.18	Balma-Gramont
83	A	0:00:00	10:59:00	1:21:00	89.05	164.75	329.5	100.0	Balma-Gramont
73	B	1:38:00	9:32:00	0:49:00	79.55	179.75	322.75	79.55	Borderouge
73	26	1:12:00	9:58:00	0:46:00	83.52	184.0	333.5	81.25	Borderouge
73	36	0:23:00	10:47:00	0:55:00	89.24	179.0	340.75	90.36	Borderouge
73	41	6:28:00	4:42:00	0:30:00	40.29	93.0	163.5	75.81	Borderouge
111	79	1:13:00	9:24:00	1:02:00	80.69	187.5	328.5	75.2	Ramonville Métro

A. THE THROUGHPUT IMPROVEMENTS BY NC

line1	line2	t_1	t_{12}	t_2	$r(\%)$	D(GB)	$D_{nc}(GB)$	$G_t(\%)$	Station
111	B	1:14:00	9:23:00	0:58:00	81.01	184.25	325.0	76.39	Ramonville Métro
111	112	2:30:00	8:07:00	0:29:00	73.12	143.5	265.25	84.84	Ramonville Métro
111	88	0:15:00	10:22:00	1:13:00	87.61	166.75	322.25	93.25	Ramonville Métro
111	68	1:00:00	9:37:00	1:02:00	82.55	189.25	333.5	76.22	Ramonville Métro
111	37	0:34:00	10:03:00	1:27:00	83.29	176.25	327.0	85.53	Ramonville Métro
111	62	0:06:00	10:31:00	1:35:00	86.2	162.25	320.0	97.23	Ramonville Métro
109	202	0:37:00	9:27:00	1:19:00	83.02	169.5	311.25	83.63	Castanet-Tolosan
109	62	0:31:00	9:33:00	1:50:00	80.25	166.5	309.75	86.04	Castanet-Tolosan
109	205	0:45:00	9:19:00	1:19:00	81.84	173.5	313.25	80.55	Castanet-Tolosan
201	79	0:16:00	10:16:00	1:05:00	88.38	166.0	320.0	92.77	Saint Orens Lycée
201	78	0:09:00	10:23:00	1:44:00	84.65	162.5	318.25	95.85	Saint Orens Lycée
79	B	1:15:00	9:11:00	1:10:00	79.17	190.25	328.0	72.4	Ramonville Métro
79	78	0:09:00	11:12:00	0:55:00	91.3	174.75	342.75	96.14	Saint Orens Lycée
79	112	2:17:00	8:09:00	0:27:00	74.89	142.5	264.75	85.79	Ramonville Métro
79	88	0:27:00	9:59:00	1:36:00	82.96	170.0	319.75	88.09	Ramonville Métro
79	68	0:53:00	9:33:00	1:06:00	82.8	183.0	326.25	78.28	Ramonville Métro
79	37	0:22:00	10:04:00	1:26:00	84.83	167.5	318.5	90.15	Ramonville Métro
79	62	0:07:00	10:19:00	1:47:00	84.45	160.0	314.75	96.72	Ramonville Métro
B	26	1:16:00	9:05:00	1:39:00	75.69	193.25	329.5	70.5	Borderouge
B	112	2:18:00	8:03:00	0:33:00	73.85	145.5	266.25	82.99	Ramonville Métro
B	88	0:27:00	9:54:00	1:41:00	82.27	168.75	317.25	88.0	Ramonville Métro
B	68	1:04:00	9:17:00	1:22:00	79.23	187.25	326.5	74.37	Ramonville Métro
B	37	0:38:00	9:43:00	1:47:00	80.08	174.25	320.0	83.64	Ramonville Métro
B	36	0:26:00	9:55:00	1:47:00	81.73	168.25	317.0	88.41	Borderouge
B	62	0:07:00	10:14:00	1:52:00	83.77	158.75	312.25	96.69	Ramonville Métro
B	41	6:03:00	4:18:00	0:54:00	38.22	105.0	169.5	61.43	Borderouge
116	48	0:07:00	9:07:00	1:32:00	84.67	142.0	278.75	96.3	Tournefeuille Lycée
11	49	0:50:00	9:29:00	1:10:00	82.58	179.75	322.0	79.14	Basso Cambo

A. THE THROUGHPUT IMPROVEMENTS BY NC

line1	line2	t_1	t_{12}	t_2	$r(\%)$	D(GB)	$D_{nc}(GB)$	$G_t(\%)$	Station
11	53	0:52:00	9:27:00	1:07:00	82.65	180.75	322.5	78.42	Basso Cambo
11	38	0:32:00	9:49:00	1:05:00	85.86	171.25	318.5	85.99	Empalot
11	14	0:17:00	10:02:00	1:49:00	82.69	163.25	313.75	92.19	Basso Cambo
11	48	6:06:00	4:13:00	0:52:00	37.7	102.25	165.5	61.86	Basso Cambo
11	54	0:20:00	10:01:00	1:05:00	87.61	165.25	315.5	90.92	Empalot
11	A	0:00:00	10:19:00	2:01:00	83.65	154.75	309.5	100.0	Basso Cambo
11	57	1:48:00	8:31:00	1:14:00	73.74	183.25	311.0	69.71	Basso Cambo
11	58	1:29:00	8:50:00	0:47:00	79.58	167.75	300.25	78.99	Basso Cambo
11	8	1:48:00	8:31:00	1:37:00	71.37	200.5	328.25	63.72	Basso Cambo
22	14	0:25:00	11:45:00	0:06:00	95.79	180.75	357.0	97.51	Marengo-SNCF
26	36	0:21:00	10:23:00	1:19:00	86.17	171.5	327.25	90.82	Borderouge
26	41	6:15:00	4:29:00	0:43:00	39.16	99.5	166.75	67.59	Borderouge
78	56	1:57:00	9:28:00	0:25:00	80.0	160.75	302.75	88.34	Université Paul Sabatier
78	115	2:27:00	8:58:00	0:47:00	73.5	169.75	304.25	79.23	Université Paul Sabatier
78	34	1:39:00	9:46:00	0:07:00	84.68	151.75	298.25	96.54	Université Paul Sabatier
78	82	9:46:00	1:39:00	0:16:00	14.12	36.75	61.5	67.35	Université Paul Sabatier
56	115	1:26:00	8:27:00	1:18:00	75.56	185.25	312.0	68.42	Université Paul Sabatier
56	34	1:19:00	8:34:00	1:19:00	76.49	187.75	316.25	68.44	Université Paul Sabatier
56	82	8:32:00	1:21:00	0:34:00	12.92	45.75	66.0	44.26	Université Paul Sabatier
112	88	0:10:00	8:26:00	3:09:00	71.77	134.0	260.5	94.4	Ramonville Métro
112	68	0:32:00	8:04:00	2:35:00	72.13	145.0	266.0	83.45	Ramonville Métro
112	37	0:10:00	8:26:00	3:04:00	72.29	134.0	260.5	94.4	Ramonville Métro
112	62	0:05:00	8:31:00	3:35:00	69.9	131.5	259.25	97.15	Ramonville Métro
65	34	0:48:00	10:22:00	1:03:00	84.86	191.5	347.0	81.2	Arènes
49	53	1:09:00	9:30:00	1:04:00	81.08	190.5	333.0	74.8	Basso Cambo
49	14	0:27:00	10:12:00	1:39:00	82.93	173.25	326.25	88.31	Basso Cambo
49	48	5:59:00	4:40:00	0:25:00	42.17	88.75	158.75	78.87	Basso Cambo
49	A	0:00:00	10:39:00	1:41:00	86.35	159.75	319.5	100.0	Basso Cambo

line1	line2	t_1	t_{12}	t_2	$r(\%)$	D(GB)	$D_{nc}(GB)$	$G_t(\%)$	Station
49	57	1:57:00	8:42:00	1:03:00	74.36	177.75	308.25	73.42	Basso Cambo
49	58	1:39:00	9:00:00	0:37:00	79.88	162.75	297.75	82.95	Basso Cambo
49	8	1:36:00	9:03:00	1:05:00	77.13	184.5	320.25	73.58	Basso Cambo
115	34	1:32:00	8:13:00	1:40:00	71.97	192.25	315.5	64.11	Université Paul Sabatier
115	82	8:04:00	1:41:00	0:14:00	16.86	35.75	61.0	70.63	Université Paul Sabatier
20	51	3:29:00	7:28:00	0:10:00	67.17	119.5	231.5	93.72	Balma-Gramont
20	77	3:50:00	7:07:00	0:46:00	60.74	141.25	248.0	75.58	Balma-Gramont
20	A	0:00:00	10:57:00	1:23:00	88.78	164.25	328.5	100.0	Balma-Gramont
51	68	0:54:00	1:16:00	6:21:00	14.87	59.5	78.5	31.93	La Terrasse
51	77	2:31:00	5:07:00	2:46:00	49.2	190.0	266.75	40.39	Balma-Gramont
51	A	0:00:00	7:38:00	4:42:00	61.89	114.5	229.0	100.0	Balma-Gramont
88	68	1:11:00	10:24:00	0:15:00	87.89	167.25	323.25	93.27	Ramonville Métro
88	37	0:29:00	11:06:00	0:24:00	92.63	184.5	351.0	90.24	Ramonville Métro
88	62	0:07:00	11:28:00	0:38:00	93.86	177.25	349.25	97.04	Ramonville Métro
53	14	0:13:00	10:21:00	1:30:00	85.77	165.0	320.25	94.09	Basso Cambo
53	48	5:59:00	4:35:00	0:30:00	41.42	91.25	160.0	75.34	Basso Cambo
53	A	0:00:00	10:34:00	1:46:00	85.68	158.5	317.0	100.0	Basso Cambo
53	57	1:50:00	8:44:00	1:01:00	75.4	176.75	307.75	74.12	Basso Cambo
53	58	1:44:00	8:50:00	0:47:00	77.83	167.75	300.25	78.99	Basso Cambo
53	8	1:53:00	8:41:00	1:27:00	72.26	195.5	325.75	66.62	Basso Cambo
68	37	0:25:00	10:14:00	1:16:00	85.87	172.25	325.75	89.11	Ramonville Métro
68	62	0:00:00	10:39:00	1:27:00	88.02	159.75	319.5	100.0	Ramonville Métro
37	62	0:09:00	11:21:00	0:45:00	92.65	177.0	347.25	96.19	Ramonville Métro
77	A	0:00:00	7:53:00	4:27:00	63.92	118.25	236.5	100.0	Balma-Gramont
36	41	6:41:00	5:01:00	0:11:00	42.22	83.5	158.75	90.12	Borderouge
34	82	8:37:00	1:16:00	0:39:00	12.03	48.25	67.25	39.38	Université Paul Sabatier
202	62	0:44:00	10:02:00	1:21:00	82.81	183.5	334.0	82.02	Castanet-Tolosan
202	205	0:09:00	10:37:00	0:01:00	98.45	160.0	319.25	99.53	Castanet-Tolosan

A. THE THROUGHPUT IMPROVEMENTS BY NC

line1	line2	t_1	t_{12}	t_2	r(%)	D(GB)	$D_{nc}(GB)$	G_t(%)	Station
62	205	1:25:00	9:58:00	0:40:00	82.71	179.5	329.0	83.29	Castanet-Tolosan
38	54	0:49:00	10:05:00	1:01:00	84.62	188.0	339.25	80.45	Empalot
14	48	6:57:00	4:54:00	0:11:00	40.72	81.75	155.25	89.91	Basso Cambo
14	A	0:00:00	11:51:00	0:29:00	96.08	177.75	355.5	100.0	Basso Cambo
14	57	2:15:00	9:36:00	0:09:00	80.0	150.75	294.75	95.52	Basso Cambo
14	58	2:30:00	9:21:00	0:16:00	77.17	152.25	292.5	92.12	Basso Cambo
14	8	2:10:00	9:41:00	0:27:00	78.73	165.5	310.75	87.76	Basso Cambo
87	8	0:38:00	10:48:00	0:50:00	88.04	190.5	352.5	85.04	Cité Scolaire Rive Gauche
48	A	0:00:00	5:05:00	7:15:00	41.22	76.25	152.5	100.0	Basso Cambo
48	57	1:14:00	3:51:00	5:54:00	35.05	113.25	171.0	50.99	Basso Cambo
48	58	1:19:00	3:46:00	5:51:00	34.45	115.75	172.25	48.81	Basso Cambo
48	8	0:40:00	4:25:00	5:43:00	40.9	96.25	162.5	68.83	Basso Cambo
A	57	2:35:00	9:45:00	0:00:00	79.05	146.25	292.5	100.0	Basso Cambo
A	58	2:43:00	9:37:00	0:00:00	77.97	144.25	288.5	100.0	Basso Cambo
A	8	2:12:00	10:08:00	0:00:00	82.16	152.0	304.0	100.0	Basso Cambo
57	58	1:23:00	8:22:00	1:15:00	76.06	181.75	307.25	69.05	Basso Cambo
57	8	1:25:00	8:20:00	1:48:00	72.15	188.75	313.75	66.23	Basso Cambo
58	8	1:34:00	8:03:00	2:05:00	68.8	191.25	312.0	63.14	Basso Cambo
In total:		-	-	-	71.58	22325.25	40762.5	82.58	-

Publications

- Qiankun Su, Katia Jaffrès-Runser, Gentian Jakllari and Charly Poulliat, "**An efficient content delivery infrastructure leveraging the public transportation network**," in *19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'16)*, , Malta, Nov. 2016. doi: 10.1145/2988287.2989152.
- Qiankun Su, Katia Jaffrès-Runser, Gentian Jakllari and Charly Poulliat, "**Xor network coding for data mule delay tolerant networks**", in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, 2015, pp. 1–6. doi: 10.1109/ICCChina.2015.7448634. (invited)

Bibliography

[1] M. Rai, N. Garg, S. Verma, and S. Tapaswi, "A new heuristic approach for minimum connected dominating set in adhoc wireless networks," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*. IEEE, 2009, pp. 284–289.

[2] (https://esa.un.org/unpd/wup) United Nations, World Urbanization Prospects: The 2014 Revision. [Online]. Available: https://esa.un.org/unpd/wup/

[3] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: Definitions, dimensions, performance, and initiatives," *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.

[4] V. N. I. Cisco, "Global mobile data traffic forecast update, 2015–2020," *white paper*, Feb 2016.

[5] A. Kumar, D. Y. Liu, and J. Sengupta, "Divya ,Äúevolution of mobile wireless communication networks: 1g to 4g.,Äù," *International Journal of electronics & communication technology*, vol. 1, no. 1, pp. 68–72, 2010.

[6] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[7] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5g of terrestrial mobile telecommunication," *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 36–43, 2014.

[8] B. Baron, P. Spathis, H. Rivano, and M. D. de Amorim, "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3248–3261, 2016.

[9] A. S. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[10] K. Jaffres-Runser and C. Lauradoux, "Authentication planning for xor network coding," in *Network Coding (NetCod), 2011 International Symposium on*. IEEE, 2011, pp. 1–6.

[11] Q. Su, K. Jaffrès-Runser, G. Jakllari, and C. Poulliat, "An efficient content delivery infrastructure leveraging the public transportation network," in *19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'16)*, , Malta, Nov. 2016.

[12] Q. Su, K. Jaffres-Runser, G. Jakllari, and C. Poulliat, "Xor network coding for data mule delay tolerant networks," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Nov 2015, pp. 1–6.

[13] R. Murtagh, "Mobile now exceeds pc: The biggest shift since the internet began," *Search Engine Watch*, vol. 8, 2014.

[14] A.-S. K. Pathan, M. M. Monowar, and S. Khan, *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test*. CRC Press, 2014.

[15] F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.

[16] H. Claussen, L. T. Ho, and L. G. Samuel, "An overview of the femtocell concept," *Bell Labs Technical Journal*, vol. 13, no. 1, pp. 221–245, 2008.

[17] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications magazine*, vol. 46, no. 9, pp. 59–67, 2008.

[18] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 104–112, 2013.

[19] L. Cui and M. B. H. Weiss, "Can unlicensed bands be used by unlicensed usage?" in *TPRC 41: The 41st Research Conference on Communication, Information and Internet Policy.*, March 2013.

[20] E. Bulut and B. K. Szymanski, "Wifi access point deployment for efficient mobile data offloading," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 17, no. 1, pp. 71–78, 2013.

[21] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 2, pp. 536–550, 2013.

[22] J. Lee, Y. Yi, S. Chong, and Y. Jin, "Economics of wifi offloading: Trading delay for cellular capacity," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1540–1554, 2014.

[23] X. Kang, Y.-K. Chia, S. Sun, and H. F. Chong, "Mobile data offloading through a third-party wifi access point: an operator's perspective," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5340–5351, 2014.

[24] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[25] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Coding or not: Optimal mobile data offloading in opportunistic vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 318–333, 2014.

[26] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of network and computer applications*, vol. 37, pp. 380–392, 2014.

[27] J. Chen, B. Liu, L. Gui, F. Sun, and H. Zhou, "Engineering link utilization in cellular offloading oriented vanets," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[28] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using wifi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 199–210.

[29] A. Pentland, R. Fletcher, and A. A. Hasson, "A road to universal broadband connectivity," in *Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Innovation (dyd 02), Bangalore, India*, 2002.

[30] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *INFOCOM*, vol. 6, 2006, pp. 1–11.

[31] M. R. Schurgot, C. Comaniciu, and K. Jaffres-Runser, "Beyond traditional dtn routing: social networks for opportunistic communication," *IEEE Communications Magazine, Volume 50, Number 7, July 2012*, vol. 50, no. 7, pp. 155–162, 2013.

[32] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[33] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*. ACM, 2004, vol. 34, no. 4.

[34] Delay-tolerant networking research group (dtnrg). [Online]. Available: https://irtf.org/concluded/dtnrg

[35] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 260–267.

[36] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[37] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," IETF, Tech. Rep., 2007, rFC 4838.

[38] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, May 2011.

[39] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[40] P. Unwin, *ICT4D: Information and communication technology for development*. Cambridge University Press, 2009.

[41] A. V. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay tolerant networks: Protocols and applications*. CRC press, 2016.

[42] E. A. Brewer, "A scalable enabling it infrastructure for developing regions (ict4b)," *NSF Proposal*, 2003.

[43] M. Demmer, B. Du, and M. Piotrowski, "E-mail4b: An e-mail system for the developing world," 2008.

[44] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," IETF, Tech. Rep., 2003.

[45] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*. Springer, 1996, pp. 153–181.

[46] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[47] T. Spyropoulos, K. Psounis, and C. 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*. ACM, 2005, pp. 252–259.

[48] Y. Cao and Z. Sun, "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 654–677, 2013.

[49] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[50] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[51] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.

[52] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE transactions on information theory*, vol. 49, no. 2, pp. 371–381, 2003.

[53] M. Médard and A. Sprintson, *Network coding: fundamentals and applications*. Academic Press, 2011.

[54] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 3, pp. 497–510, 2008.

[55] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782–795, 2003.

[56] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Information Theory, 2003. Proceedings. IEEE International Symposium on*. IEEE, 2003, p. 442.

[57] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[58] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 41, no. 1. Citeseer, 2003, pp. 11–20.

[59] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: Opportunities & challenges," in *MILCOM 2007-IEEE Military Communications Conference*. IEEE, 2007, pp. 1–8.

[60] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 284–291.

[61] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, "Benefits of network coding for unicast application in disruption-tolerant networks," *Networking, IEEE/ACM Transactions on*, vol. 21, no. 5, pp. 1407–1420, 2013.

[62] N. Shrestha and L. Sassatelli, "Inter-session network coding-based policies for delay tolerant mobile social networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7329–7342, 2016.

[63] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "More: A network coding approach to opportunistic routing," Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, Tech. Rep., 2006.

[64] A. Khreishah, W. Chih-Chun, and N. B. Shroff, "Rate control with pairwise intersession network coding," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 816–829, June 2010.

[65] M. Heindlmaier, D. Lun, D. Traskov, and M. Medard, "Wireless inter-session network coding - an approach using virtual multicasts," in *IEEE International Conference on Communication*, 2011, pp. 1–5.

[66] S. Ahmed and S. S. Kanhere, "Hubcode: hub-based forwarding using network coding in delay tolerant networks," *Wireless Communication and Mobile Computation*, vol. 13, no. 9, May 2011.

[67] N. Shrestha and L. Sassatelli, "Inter-session network coding in delay tolerant mobile social networks: An empirical study," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a.* IEEE, 2015, pp. 1–6.

[68] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 2, pp. 450–463, 2008.

[69] Google. The general transit feed specification (gtfs). [Online]. Available: https://developers.google.com/transit/gtfs/

[70] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, 2009, p. 55.

[71] E. Sampathkumar and H. B. Walikar, "The connected domination number of a graph," *J. Math. Phys. Sci*, vol. 13, no. 6, pp. 607–613, 1979.

[72] M. R. Garey and D. S. Johnson, "A guide to the theory of np-completeness," *WH Freemann, New York*, 1979.

[73] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected dominating set in sensor networks and manets," in *Handbook of combinatorial optimization*. Springer, 2004, pp. 329–369.

[74] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks–a comprehensive survey," *Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013.

[75] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 376–380.

[76] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 7, pp. 721–730, 2007.

[77] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[78] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[79] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1979.

[80] L. C. Freeman, S. P. Borgatti, and D. R. White, "Centrality in valued graphs: A measure of betweenness based on network flow," *Social networks*, vol. 13, no. 2, pp. 141–154, 1991.

[81] M. E. Newman, "A measure of betweenness centrality based on random walks," *Social networks*, vol. 27, no. 1, pp. 39–54, 2005.

[82] U. Brandes and D. Fleischer, *Centrality measures based on current flow*. Springer, 2005.

[83] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," Stanford InfoLab, Tech. Rep., 1999.