An Efficient Content Delivery Infrastructure Leveraging the Public Transportation Network

Qiankun Su
IRIT-INPT/ENSEEIHT
University of Toulouse
Toulouse, France
qiankun.su@enseeiht.fr

Katia Jaffrès-Runser
IRIT-INPT/ENSEEIHT
University of Toulouse
Toulouse, France
kjr@enseeiht.fr

Gentian Jakllari
IRIT-INPT/ENSEEIHT
University of Toulouse
Toulouse, France
jakllari@enseeiht.fr

Charly Poulliat
IRIT-INPT/ENSEEIHT
University of Toulouse
Toulouse, France
charly.poulliat@enseeiht.fr

ABSTRACT

With the world population becoming increasingly urban and the multiplication of mega cities, urban leaders have responded with plans calling for so called smart cities relying on instantaneous access to information using mobile devices for an intelligent management of resources. Coupled with the advent of the smartphone as the main platform for accessing the Internet, this has created the conditions for the looming wireless bandwidth crunch.

This paper presents a content delivery infrastructure relying on off-the-shelf technology and the public transportation network (PTN) aimed at relieving the wireless bandwidth crunch in urban centers. Our solution proposes installing WiFi access points on selected public bus stations and buses and using the latter as data mules, creating a delay tolerant network capable of carrying content users can access while using the public transportation. Building such an infrastructure poses several challenges, including congestion points in major hubs and the cost of additional hardware necessary for secure communications. To address these challenges we propose a 3-Tier architecture that guarantees end-to-end delivery and minimizes hardware cost. Trace-based simulations from three major European cities of Paris, Helsinki and Toulouse 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 several times while delivering at least as many packets as a baseline architecture.

Keywords

public transportation networks; XOR network coding; content delivery; urban data offloading, smart cities.

MSWiM '16, November 13-17, 2016, Malta, Malta
© 2016 ACM. ISBN 978-1-4503-4502-6/16/11...$15.00
DOI: http://dx.doi.org/10.1145/2988287.2989152

1. 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 [1]. Cities are responding to the environmental, transportation and infrastructure challenges this poses by becoming more intelligent, interconnected and efficient, making information and communications technologies (ICTs) crucial 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 a annual growth rate of 53% between 2015 and 2020, reaching 30.6 exabytes per month by 2020 [8]. 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 paper, 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 proposal is based on the observation that a significant part of the mobile content is consumed in urban areas while people are commuting using public transportation. The solution, depicted in Figure 1, proposes to install WiFi access points on buses and bus stations. The buses act as data mules creating a data mule delay tolerant network capable of carrying content that 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 in turn publish new content.

The main advantage of the proposed content delivery infrastructure is that it relies on inexpensive WiFi technology and an extensive public transportation network already in place. However, public transportation networks are built around the concept of hubs with many bus lines converging on few major stops around city centers. As a result, the

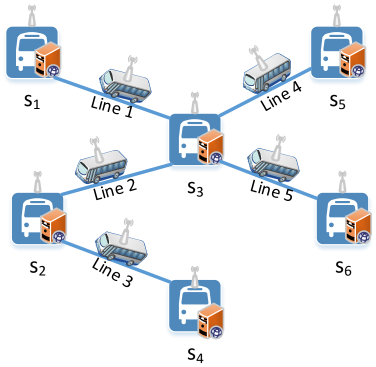


Figure 1: Content delivery using PTNs.

network can suffer from congestion points, such as the one shown in Figure 1. Four buses are shown to converge on stop S_3 and in the short time of a normal stop they all need to upload traffic to the bus station access point who in turn needs to push traffic to all four buses. Medium access control protocols, including IEEE 802.11 MAC, are designed to share the channel fairly among stations, leading to a situation where S_3 needs to communicate to all four buses but gets only a fifth of the channel capacity.

To address the challenge of congestion points, we propose to use network coding, which has been shown to significantly improve the system throughput in such scenarios [11]. Relay nodes, such as the station S_3, linearly combine sets of messages into a new network coded message (i.e. encode). Typically, a coded message m_c is obtained as the weighted sum of a set of K messages $m_{i,i\in[1..K]}$ following $m_c = \sum_{i=1}^{K} \alpha_i m_i$, where the coefficient $\alpha_i, i = 1 \cdots K$, are randomly chosen scalar elements from a finite field $\mathbb{F}_q$. In this paper, we consider basic XOR-network coding where messages are simply xor-ed together: $m_c^{xor} = \oplus_{i=1}^{K} m_i$. The destination nodes, the four buses in our example, extract the desired messages (i.e. decode) by solving the linear system created by these coded messages and their coefficients. However, network coding is suffering from a new security thread, naming pollution attacks. It spreads the pollution by combining legitimate messages with polluted ones and therefore limiting the recovery probability of legitimate messages [10]. It is possible to mitigate pollution attacks with advanced message authentication strategies but at the cost of complex cryptographic operations [4] requiring sophisticated and more expensive hardware.

To address the challenge arising from constructing a secure and cost-effective platform, we introduce an architecture that classifies bus stops in PTNs into 3 tiers. The first tier groups the biggest number of bus stops and on which we will not install any wireless access points. The second biggest group consists of bus stops on which we install basic wireless access points that are capable of relaying data traffic but cannot implement secure network coding. Finally, to minimize cost, a very limited number of bus stations belonging to the third tier will be equipped with powerful access points implementing secure network coding. Implementing such architecture, however, introduces several challenges, chief among them, how to assign the bus stops into every tier such that end-to-end connectivity in guaranteed, packet delivery is maximized and the infrastructure cost is minimized. In short, we address these challenges by using a constructive method. First, we identify bus stops that do not need to relay any data, making them good candidates for the first tier. Second, we populate the second tier by identifying the minimum number of bus stops for guaranteeing end-to-end connectivity. Finally, among the bus stops from the second tier, we select a very small subset, judged to be the most important, to be included in the third tier.

Throughout this paper we make the following contributions: *i*) in Section 3, we introduce the content delivery infrastructure leveraging PTNs ; *ii*) in Section 4, we introduce our solution for using network coding and provide bounds on its throughput gains ; *iii*) in Section 5, we introduce a 3-Tier architecture for a secure and cost effective content delivery using PTNs and *iv*) in Section 6, we use real traces and bus schedules from three major European cities to evaluate the performance of our content delivery infrastructure.

2. RELATED WORK

Leveraging public transportation to carry delay tolerant data has been proposed in the past. Main motivations were to offer connectivity to remote locations in underdeveloped countries or to provide a disaster-relief communication infrastructure. Real deployments have been tested. DakNet [15] provides a low-cost Internet access to remote villages in India and Cambodia. The UMass DieselNet [6] has equipped up to 40 buses with access points in Amherst where data is delivered by bus-to-bus communications, leveraging their intermittent connectivity. This work has led to the design of the well known MaxProp routing protocol. 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.

Routing data in delay tolerant networks (DTN) usually leverages intermittent contacts between mobile entities using the so-called "store-carry-forward" paradigm. Transfer of data is done on a per-encounter basis: two nodes carry content and exchange it upon contact such as to rapidly diffuse a copy of it to the destination. This baseline protocol is the epidemic protocol [22], which is known to be way too resource consuming due to its elevated replication rate. Main DTN routing protocols have been designed for entities that exhibit a non-predictable mobility pattern [18]. In our case, the mobility of buses is predictable and thus, leading to more efficient routing solutions [6] where vehicle to vehicle communications are exploited to create a village communication 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 two access points. Routing paths can be calculated a priori knowing the bus line topology, final transfer being adjusted to actual arrival and departures dates of buses on the fly.

As pinpointed earlier, our content delivery infrastructure usually holds several points of congestion that can be relieved by network coding [2]. Network coding solutions for

DTNs were mostly discussed in the context of people-centric networks [9, 13, 17, 19, 23], where mobility of nodes is non-predictable. In these solutions, intra- or inter-flow network coding techniques have been proposed to route end-to-end flows. Messages are encoded by source or relay nodes and generally only decoded by the destination node. Complex decisions have to be made to select the messages to encode together such as to maximise the probability of decoding at the destination. Recent solutions have leveraged social routing information to improve the coding procedure at relays [19], but at the cost of memorizing the full social graph information captured by the protocol in the nodes.

On our side, we have opted for a solution where hop-by-hop encoding and decoding is performed using simple XOR operations. This approach has been successfully implemented in practice for wireless mesh networks in COPE [11]. This scheme is less effective for DTNs than for wireless mesh networks because it is rarely possible to overhear neighbors' transmissions [17]. However, it is still possible to apply the XOR scheme of COPE to pairwise communications crossing at a node of the network [13]. Exploiting this feature, we have demonstrated theoretically in previous works [20] that the maximum delivery probability could be increased of up to 50% in a village-to-village communication network.

To provide a cost-efficient design of our content delivery infrastructure, a reduced number of wireless access points have to be rolled out. Maintaining full connectivity with a reduced number of routing enabled nodes is possible by selecting the ones forming a connected dominating set of the graph (CDS). In this case, all non-CDS nodes push their traffic to the closest CDS node. CDS is then in charge of routing data [21] to the destination. In our paper, we leverage CDS to select the stations where access points are rolled out. Selecting the minimum subset of nodes forming a connected dominating set cannot be solved in polynomial time. As such, we build our solution on a previously proposed heuristic [16] whose main idea is to form a connected dominating set by traversing all nodes (either with a breadth first search or depth first search), beginning with the node with the highest degree, and continuously removing the node v if $G(V - \{v\})$ is still connected. This heuristic is modified by accounting for different centrality measures of the graph such as betweenness [5] and page rank [14]. Next, we present in details our contributions, starting with the global architecture description of our content delivery infrastructure.

3. PTN*s* FOR CONTENT DELIVERY

3.1 Overall architecture

The content delivery infrastructure envisioned in this paper leverages public transportation networks where public vehicles such as buses act as data mules. Data mules carry content that PTN customers can access over wireless (e.g. WiFi) on the bus or waiting at selected stations. Therefore, 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.

Our 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. We count the duration of inter stops for the public bus network of the city of Paris using publicly available traces to estimate the contact time between buses and intermediate stops since the contact time is not recorded in the trace. The statistics are plotted in Figure 3, showing the duration of almost all inter stops is within 4 minutes, not to mention the contact time between the bus and intermediate stops. In contrast, Figure 3 shows that around 40% of the buses wait at the end stations for more than 10 minutes. End stations are thus particularly interesting elements of the PTN. At end stations, conductors usually rest for an extended period of time before engaging into the next journey. Moreover, multiple bus lines usually cross at such stations. This results into an extended contact duration between bus lines at end stations. We have shown in [20] that it is beneficiary to leverage such extended contact duration for exchanging data between the buses. Thus, in our architecture, we only deploy access points at the final stops of bus lines, and do not consider intermediary stops for such purpose.

City	Nodes	NLN	Edges	Buses
Toulouse	44	16	46	297
Paris	213	99	236	3056
Helsinki	217	90	266	1512

Table 1: Investigated PTN topologies

Term	Description
NC	network coding
NLN	non leaf nodes, $deg(v) \neq 1$
BW	betweenness centrality
CDS	connected dominating set
$G(V', E')$	a graph induced by CDS
Δ	message creation period, $\Delta > 0$
$bw(v)$	the betweenness centrality of the node v

Table 2: Notations and definitions

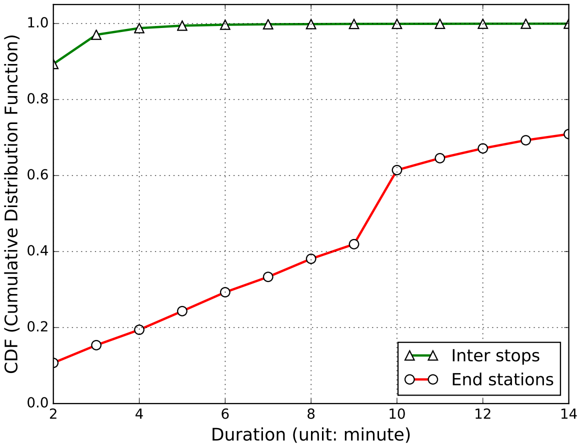
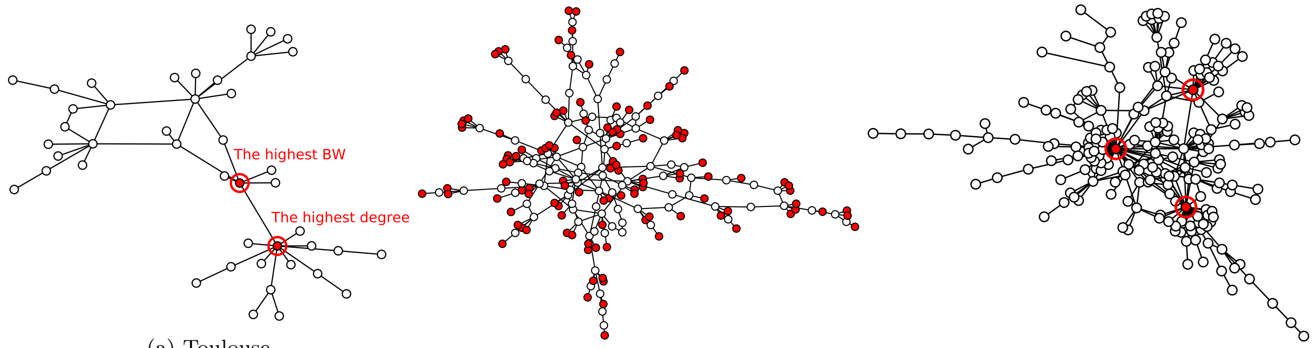


Figure 3: The CDF of the duration of inter stops and waiting time at end stations in Paris.

3.2 Models and assumptions



(a) Toulouse

(b) Paris (leaf nodes in red)

(c) Helsinki (the 3 highest PageRank in red)

Figure 2: The biggest connected component of public transportation networks.

PTN model and topologies.

The content delivery infrastructure can be modeled 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. A bus b_k of the PTN is associated with an edge (s_i, s_j).

In this paper, 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, showing the largest connected components of three different PTNs serving 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. Table 1 lists the main characteristics of these networks which have been extracted from the open dataset of PTN providers[1] in GTFS[2]. Thanks to the popularity of GTFS, our scheme is easy to adapt to other cities.

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

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 node 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.

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.

3.3 Underlying protocols

DTN routing.

From the connected graph G, we are able to pre-calculate routing tables for each station by using Dijkstra's shortest path algorithm. The routing metric to minimize is the basic hop count metric. 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 composing G.

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, the station extracts m's destination s_j and looks up its next hop s_k in the routing table. 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. From the list of buses, it extracts the set of next hop nodes 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.

Medium access control.

To be fair, we assume a medium access control at the base station that divides the bandwidth equally between the contending nodes. Thus, any fair medium access control mechanism such as CSMA or TDMA can be implemented in practice.

As such, if N buses are connected to a station, $N + 1$ emitters are concurrently contending for the communication bandwidth. In other words, a station only gets a $1/(N + 1)$

---

[1] 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

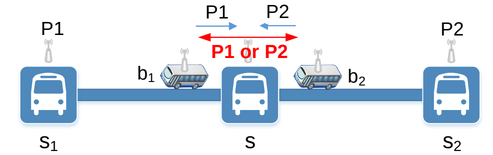
[2] GTFS (General Transit Feed Specification) is a format for public transportation schedules and associated information

share of the bandwidth at the MAC layer. However, to drain N messages coming from the N connected buses on the uplink, the station's AP has to send N messages on the downlink as well. But it only has a $1/(N+1)$ bandwidth share, while it has N times more data to send than a bus. Such an imbalance results in a significant drop in throughput under heavy traffic conditions.

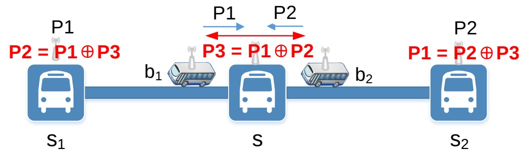
4. NETWORK CODING FOR PTN*S*

4.1 Motivation

It is possible to mitigate the unbalanced bandwidth demand by leveraging inter-session XOR network coding at stations as underlined in [20]. Main concept is represented in Figure 4.



(a) Without network coding



(b) With XOR network coding

Figure 4: Exchange two packets via a station.

Figure 4-(a) illustrates the standard solution to exchange packets at relay nodes. It takes 4 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 b_1 or b_2. As shown in Figure 4-(b), if the station s broadcasts a coded packet $P3 = P1 \oplus P2$ in a single transmission, both buses can extract a new packet by xor-ing $P3$ with the packet they have previously sent. For instance, s_1 obtains $P2$ by calculating $P3 \oplus P1 = P2$. With XOR network coding, the base station needs a single transmission instead of two, providing the most efficient use of the underlying fair communication MAC protocol.

4.2 XOR network coding implementation

This section presents our 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 message. 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 4, $P1$ 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. Encoding is done at the station s and decoding is done at next-hop stations.

Encoding.

The pseudo-code is listed in Algorithm 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} $(i \neq j)$[3]. 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.

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

Algorithm 1 Coding procedure

```
$B = \{$Buses waiting at the station $\}$
$S = \{$Next hop nodes to be reached by $B$ $\}$
for all $s_i \in S$ do
    for all $s_j \in S, j \neq i$ do
        if $Q_{ij} \neq \varnothing$ and $Q_{ji} \neq \varnothing$ then
            $m_i$ is picked at the head of $Q_{ij}$
            $m_j$ is picked at the head of $Q_{ji}$
            return $m_c = m_i \oplus m_j$
        end if
    end for
end for
```

Decoding.

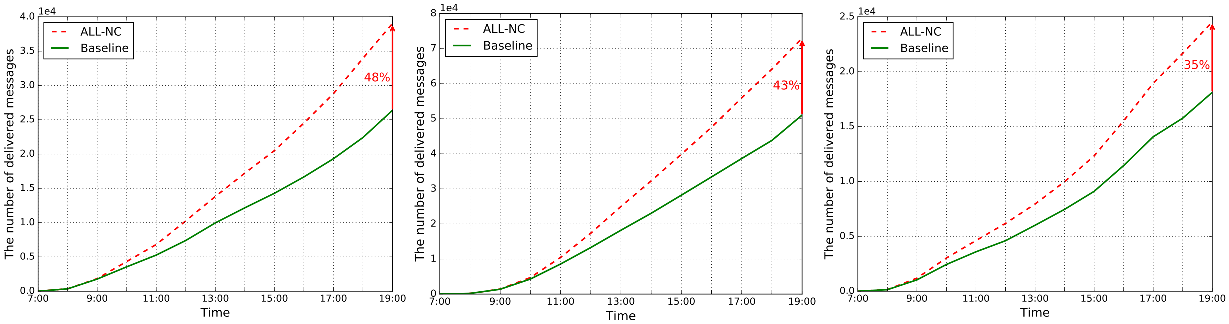
Each station keeps the messages that it has given to buses. The messages are stored in a hash table keyed on message identifier. When a station receives a XOR-ed message $m_c = m_i \oplus m_j$, it looks through the hash table to get the previously sent message, say m_i. A new message m_j is retrieved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

Bound on throughput gains.

In [20], a theoretical analysis has been conducted to derive an upper bound on the throughput gain such a XOR network coding strategy can offer for a realistic PTN. Main results are drafted to introduce the gains we can expect for the PTNs of Toulouse, Paris and Helsinki if they are leveraged for content delivery. These gains will be ascertained in this paper using fine-grained simulations in Section 6.

---

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



(a) Toulouse

(b) Paris

(c) Helsinki

Figure 5: Network coding benefit: number of delivered messages.

In our model, 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}. From this information, we can derive the amount of data that a pair of bus lines can exchange with and without network coding using Eq. (1) and Eq. (2) respectively.

$$D_{nc} = [min(t_1, t_2) + \frac{2}{3} \cdot t_{12}] \cdot R \quad (1)$$

$$D = [min(t_1, t_2) + \frac{1}{3} \cdot t_{12}] \cdot R \quad (2)$$

where R denotes the data rate. From these values, an upper bound on the throughput gain that can be expected with our pairwise inter-session XOR network coding solution can be established. Table 3 shows the overall network throughput for the Toulouse, Paris and Helsinki topologies using $R = 100Mb/s$. The expected maximum throughput gains G_t is defined as $(D_{nc}-D)/D$ and the overlapping ratio r is defined as $r_{12}/(r_1 + r_{12} + r_2)$.

The potential gain is really important and thus, XOR-network coding is a promising solution for improving the performance of our content delivery infrastructure.

This upper bound is calculated assuming all pairwise bus encounters arise without other buses being present at the same time at the station. In reality, this is not the case. Additional buses will bring more congestion to the station AP and reduce the benefit of pairwise network coding. Next, we calculate the exact XOR network-coding benefit using fine grained simulations.

4.3 Network coding benefit

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 1. Extensive simulations are carried out using the simulation setup described in Section 6.1. Two figures of merit 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.

City	r(%)	D(TB)	D_{nc}(TB)	G_t(%)
Toulouse	71.5	21	39	82.5
Paris	71.5	102	185	80.5
Helsinki	58.0	393	688	75

Table 3: The potential improvements: upper bound

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.

Figure 5 shows the number of delivered messages for baseline and ALL-NC strategies from 7 : 00 to 19 : 00. 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 (that are respectively of 82.5%, 80.5% and 75%). As already explained, this upper bound is optimistic since the pair-wise network coding can't totally compensate the traffic imbalance if more than two buses are in contact at the station.

This gain could be improved if the station could XOR more than two packets together as done in the wheel topology of [11]. But this is unfortunately not possible as the 1-hop decoding stations can't overhear the remote message emissions of the buses as done in a pure wireless setting.

Anyway, 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 6. 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

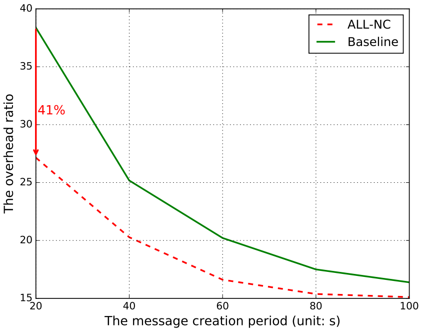


Figure 6: The overhead ratio in Paris.

XOR network coding in a single transmission instead of two with baseline. In the Paris topology, the decrease reaches 29% 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.

5. A COST-EFFICIENT AND SECURE DESIGN

Section 4.3 demonstrated the benefit of using XOR network coding in PTNs. However, before adapting it as part of our content delivery infrastructure, two main challenges have to be addressed:

1. Installing network coding enabled APs at all PTN end stations is expensive – it requires specialized hardware plus extended storage capabilities.
2. Network coding is threatened by the specific pollution attack. A malicious user could inject in the network a junk message to be XOR-ed with others. Subsequent XOR operations that include this message will pollute the rest of the network.

It is possible to mitigate pollution attacks with advanced message authentication strategies [4] but at the cost of complex cryptographic operations. Thus, installing a *secure* network coding AP is even more expensive than solely installing a network coding AP. In this section, we will concentrate on reducing the overall deployment cost such as to thwart pollution attacks. Three subsequent improvements are discussed: leaf stations removal, the 2-Tier architecture and finally the 3-Tier architecture.

5.1 Leaf stations removal

In Section 3, we proposed to leverage public transportation networks for content delivery by deploying wireless APs at all end stations. However, note that leaf stations ($deg(v) = 1$) do not need to relay any message. Thus, it is not necessary to install wireless APs at the leaf stations, leading to a drastic reduction in hardware cost. Leaf nodes are highlighted in red in Figure 2-(b) for the Paris topology. Looking at Table 1, it can save 63.5%, 53% and 52% of wireless APs in Toulouse, Paris and Helsinki, respectively. More importantly, after the leaf stations removal, the number of delivered packets in the network is unchanged – only the hardware cost is reduced.

5.2 2-Tier Architecture

The number of wireless interfaces is further reduced by calculating a connected dominating sets V' for the graph G. V' induces a connected subgraph $G'(V', E')$ of $G(V, E)$, representing a virtual backbone of the network. Only the stations belonging to V' are required to be equipped with wireless transceivers. Connected dominating sets make sure that the communication network is still connected.

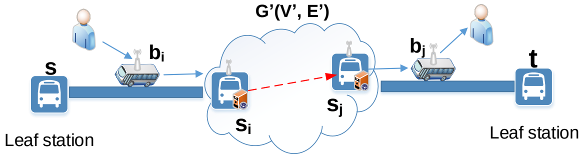


Figure 7: Relay messages in 2-tier architecture.

Figure 7 is used to demonstrate how messages are delivered in such a network. A message m is carried by a bus b_i to the station s_i ($s_i \in V'$) that dominates the leaf station s. m is relayed to s_j following the shortest path of G' and finally carried by another bus b_j to the destination t.

The main idea of MCDS-NON-DISTRIBUTED [16] is to form a connected dominating set by transversing all nodes (either breadth first search or depth first search), beginning with the node with the highest degree, and continuously removing the node v if $G(V - \{v\})$ is still connected. However, degree centrality sometimes can be deceiving for a purely local measure. Thus, we design another heuristic algorithm to construct a CDS with betweenness centrality. A connected dominating set is formed by iterating over nodes on ascending order based on betweenness centrality and continuously removing the node v if $G(V - \{v\})$ is still connected. The algorithm includes two steps: *i*), compute the shortest-path betweenness centrality for nodes V and sort them by betweenness centrality on ascending order; *ii*), transverse all nodes and continuously remove the node v if $G(V - \{v\})$ is still connected. The pseudo code is given in algorithm 2.

Algorithm 2 CDS with betweenness centrality

```
Require: A connected graph $G(V, E)$
  $d \leftarrow \{v : bw(v)\}, v \in V$, sort by BW on ascending order
  $V' \leftarrow \varnothing$, connected dominating sets
  for all $v : bw(v), v \notin V'$ do
      if $bw(v) = 0$ OR $G(V - \{v\})$ is connected then
          $V' \leftarrow V' \cup MAX - BW(N(v))$
      else
          $V' \leftarrow V' \cup \{v\}$
      end if
      $V \leftarrow V - \{v\}$
  end for
```

We first compute the shortest-path betweenness centrality for nodes V, and then sort the key-value pairs by value (i.e. betweenness centrality) on ascending order. Iterate over this sorted collection of node-betweenness pairs. A node whose betweenness centrality is equivalent to 0 (i.e. $bw(v) = 0$, v is a leaf node) can be directly removed from G. Because of this, unlike MCDS-NON-DISTRIBUTED, the leaf nodes are deleted from G without checking if the rest graph is connected, leading to improvement, especially for sparse graph. If $G(V - \{v\})$ is not connected, obviously, v must be included in the connected dominating sets V'. Otherwise, v is removed from G and the node with the highest betweenness centrality of v's neighbours is added to V'.

5.3 3-Tier Architecture

In this section, we explore how to further reduce the number of nodes performing network coding so as to improve the network security at little cost to network performance.

Motivated by the well known 80-20 rule, we examine how many nodes performing network coding are needed to achieve the performance similar to that of 2-tier architecture described in Section 5.2. Three metrics, degree, betweenness centrality and PageRank, are explored to select a subset of nodes. PageRank computes a ranking of the nodes in the graph G based on the structure of the incoming links. Since PageRank is applied to directed graphs, our undirected graph $G(V, E)$ is turned into a directed graph by assigning two directions to each edge.

Key	Value
simulation duration	12 hours
update interval	1s
time-to-live	12 hours
buffer size	infinite
message interval	20s

Table 4: Simulation parameters.

We compute the degree, betweenness centrality and PageRank for nodes in G, and then select the top n nodes that belong to CDS. The empirical results on three real traces are given in Section 6.3.

6. PERFORMANCE EVALUATION

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

i) In Section 6.2, we show that 2-Tier is able to reduce the number of wireless access points required to cover 3 major cities by approximately a factor of 3. This significant cutback in infrastructure is achieved while the packet delivery never drops below that of Baseline.

ii) In Section 6.3, we show that 3-Tier reduces the number of wireless access points capable of performing network coding required to cover 3 major cities by over an order of magnitude. 3-Tier accomplishes this cutback while delivering essentially the same performance in terms of messages delivered.

iii) In Section 6.4, we show that 3-Tier improves the cost-effectiveness by over 100% on average when compared to 2-Tier, its closest competitor.

6.1 Experimental setup

The performance evaluation are carried out on the ONE (Opportunistic Network Environment) simulator [12]. We added the broadcast mechanism to the ONE to support network coding leveraging the broadcast nature of wireless transmissions. The simulation parameters are summarized in Table 4.

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

Table 5: Number of wireless access points required to cover 3 different cities. The 2-Tier architecture reduces the required number of interfaces by approximately a factor of 3.

Real traces: 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. All traces are in GTFS (General Transit Feed Specification), developed by Google, a common format for public transportation schedules and associated geographic information. To get the mobility model, 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 bus waiting at s_i is assigned to this trip. In general, public bus services run according to schedules that are different between working days, Saturdays, Sundays and holidays. In this paper, we use working day schedules.A subset of schedules is chosen from the trace, time period ranging from 7:00 to 19:00.

City	2-Tier	3-Tier
Toulouse	13	2
Paris	85	10
Helsinki	60	3

Table 6: 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.

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

Routing: For all three topologies, route tables are generated for each router before the simulation starts so that messages are relayed based on the shortest paths.

Basis for comparison: We compare the 2-Tier and 3-Tier architectures with **Baseline**, where all nodes are equipped with a wireless access point and forward packets without performing network coding and **ALL-NC**, wherein all nodes are equipped with a wireless access point and all implement network coding.

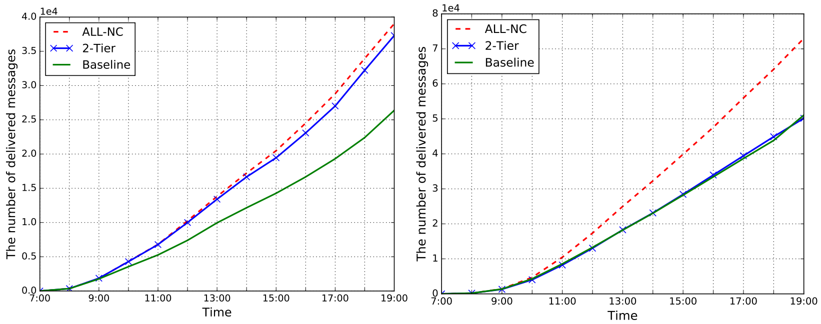
6.2 Evaluation of the 2-Tier architecture

In this section, we evaluate the 2-Tier architecture in terms of the cost of its deployment and packet delivery.

Results: Table 5 shows the number of wireless access points necessary to cover 3 major cities using the Baseline, ALL-NC and 2-Tier architectures. The results show that 2-Tier is able to reduce the number of interfaces by approximately a factor of 3. This results is the more impressive when looking into the packets delivered, shown in Figure 8. As expected, ALL-NC, equipped with over 3 times as many wireless access point and using network coding, delivers the most packets. However, the 2-Tier architecture, with a fraction of interfaces is still capable of outperforming Baseline in all 3 cities and being very competitive when compared to ALL-NC in 2 out of 3 cities.

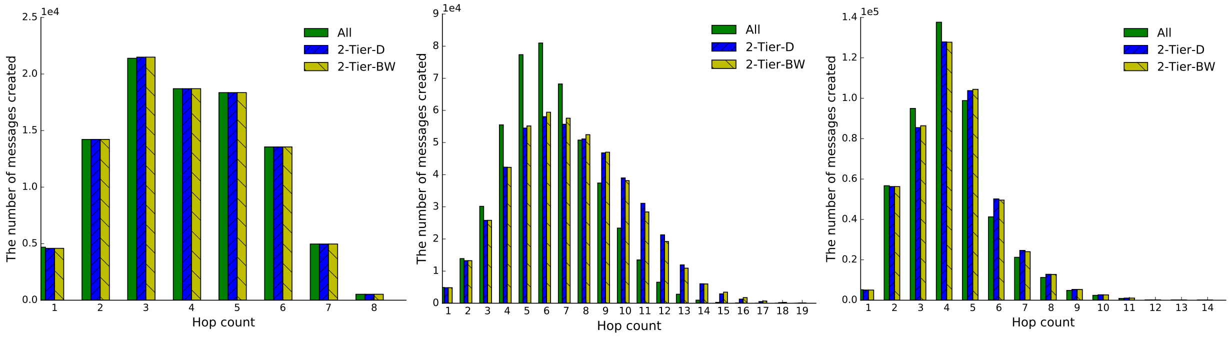
For a better understanding as to the differences in performance observed in the three cities, Figure 9 shows the average hop count of the routes utilized by all packets delivered during the simulation. Baseline and ALL-NC have the same number of access points so the routes selected are the same and shown in Figure 9 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 dramatic 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. Nevertheless, 2-Tier, thanks to network coding always outperforms Baseline despite the longer paths.

6.3 Evaluation of the 3-Tier architecture



(a) Toulouse (b) Paris (c) Helsinki

Figure 8: Number of messages delivered for Baseline, ALL-NC and 2-Tier.



(a) Toulouse (b) Paris (c) Helsinki

Figure 9: The hop count(s) of the routes taken by the messages created during the simulation.

In this section, we evaluate the performance of the 3-Tier architecture in terms of packets delivered and the number of access points deployed.

Results: Table 6 shows the number of wireless access points capable of performing network coding that 2-Tier and 3-Tier need to deploy to cover the 3 cities. The data shows that the 3-Tier architecture reduces the need for such access points by over an order of magnitude, which as we show in Section 6.4 has the potential to dramatically reduce the cost of deployment. Fortunately, this significant cutback in infrastructure does not affect performance. Figure 10 shows that 3-Tier delivers the same number of packets as 2-Tier.

6.4 Cost-effectiveness analysis

In this section, we evaluate the potential cost of different architectures for PTNs.

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 3 to a wireless access points capable of performing network coding.

Results: Figure 11 shows the cost effectiveness for all architectures. 3-Tier improves the cost effectiveness by 99.4%, 114.5%, and 115.28% for Toulouse, Paris and Helsinki, respectively over 2-Tier, its closest competitor. This validates our choice of using a 3-Tier architecture for a cost effective content delivery network leveraging the PTNs.

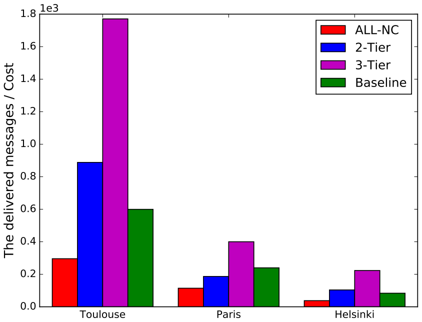
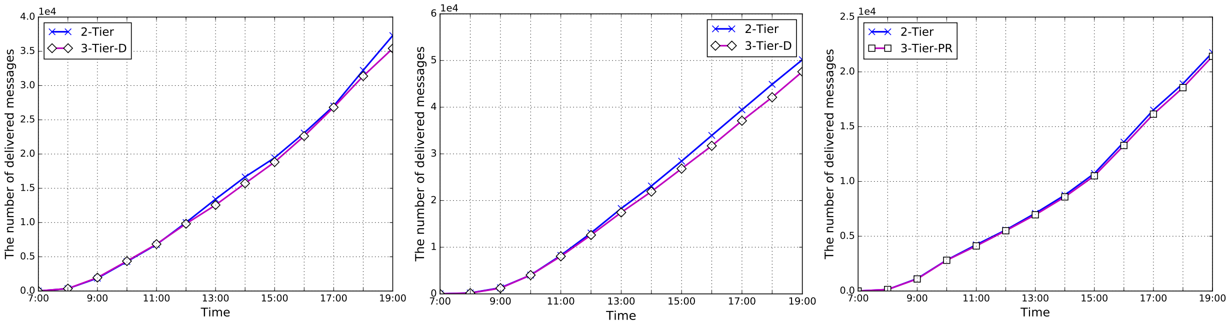


Figure 11: The cost effectiveness for all architectures.

7. CONCLUSION

We presented a secure and cost effective content delivery infrastructure leveraging existing public transportation networks aimed at relieving the looming congestion crunch in urban areas. The key novelty of our design is that it relies on inexpensive and off-the-shelf technology and infrastructure already in place. To address the challenges involved in implementing such a network, we introduced a 3-Tier architecture that is guaranteed to provide end-to-end connectivity, high packet delivery and minimizes hardware cost. We evaluated our design choices and the proposed architecture using real traces from the public transportation networks of three major European cities. The results showed that the 3-Tier architecture achieved a factor of 3 reduction in the number of access points required while delivering more



(a) Toulouse (b) Paris (c) Helsinki

Figure 10: Packets delivered for 2-Tier and 3-Tier.

messages than a baseline architecture.

Acknowledgements

This work is supported in part by CHIST-ERA MACACO project, ANR-13-CHR2-0002-06 and in part by China Scholarship Council (CSC).

8. REFERENCES

[1] United Nations, World Urbanization Prospects: The 2014 Revision, https://esa.un.org/unpd/wup.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[3] V. Albino, U. Berardi, and R. M. Dangelico. Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22(1), 2015.

[4] A. Apavatjrut, W. Znaidi, A. Fraboulet, C. Goursaud, K. Jaffrès-Runser, C. Lauradoux, and M. Minier. Energy efficient authentication strategies for network coding. *Concurrency and Computation: Practice and Experience*, 24(10):1086–1107, 2012.

[5] U. Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, 2008.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, volume 6, pages 1–11, 2006.

[7] S. Chen and J. Zhao. The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication. *Communications Magazine, IEEE*, 52(5):36–43, 2014.

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

[9] M. Heindlmaier, D. Lun, D. Traskov, and M. Medard. Wireless inter-session network coding - an approach using virtual multicasts. In *Proceedings of IEEE ICC*, pages 1–5, 2011.

[10] K. Jaffres-Runser and C. Lauradoux. Authentication planning for XOR network coding. In *Proceedings of NetCod'11*, pages 1–6. IEEE, 2011.

[11] 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)*, 16(3):497–510, 2008.

[12] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of SIMUTools'09*, page 55, 2009.

[13] A. Khreishah, W. Chih-Chun, and N. B. Shroff. Rate control with pairwise intersession network coding. *IEEE/ACM Transactions on Networking*, 18(3):816–829, June 2010.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. *Technical report. Stanford InfoLab*, 1999.

[15] A. S. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[16] M. Rai, N. Garg, S. Verma, and S. Tapaswi. A new heuristic approach for minimum connected dominating set in adhoc wireless networks. In *Proceedings of IACC 2009*, pages 284–289. IEEE, 2009.

[17] S. Ahmed and S. S. Kanhere. Hubcode: hub-based forwarding using network coding in delay tolerant networks. *Wireless Communication and Mobile Computation*, 13(9), May 2011.

[18] M. R. Schurgot, C. Comaniciu, and K. Jaffres-Runser. Beyond traditional DTN routing: social networks for opportunistic communication. *IEEE Communications Magazine*, 50(7):155–162, 2013.

[19] N. Shrestha and L. Sassatelli. Inter-session network coding in delay tolerant mobile social networks: An empirical study. In *Proceedings of IEEE WoWMoM, 2015*, pages 1–6. IEEE, 2015.

[20] Q. Su, K. Jaffrès-Runser, G. Jakllari, and C. Poulliat. XOR network coding for data mule delay tolerant networks. In *Proceedings of ICCC'15*, Nov 2015.

[21] 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*, 6(7):721–730, 2007.

[22] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

[23] 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*, 21(5):1407–1420, 2013.