Hide and Code: Session Anonymity in Wireless Line Networks with Coded Packets

Hugo Sousa-Pinto, Daniel E. Lucani and João Barros
Instituto de Telecomunicações, Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto
Emails: hugo.sousa.pinto@fe.up.pt, dlucani@fe.up.pt, jbarros@fe.up.pt

***Abstract*—The broadcast nature of the communication channel enables a malicious eavesdropper to gain information about connectivity and active sessions in a multi-hop wireless network. This can be achieved simply by overhearing the transmitted signals over the ether and analyzing their timings. Focusing on techniques that can meet information-theoretic criteria for session anonymity under traffic analysis attacks, we rely on a judicious choice of transmission schedules to conceal multicast or bidirectional unicast sessions from a global eavesdropper at any given point in time. A systematic approach for constructing the aforementioned transmission schedules for arbitrary network topologies is derived from an equivalent coloring problem in an auxiliary conflict graph. Although this type of anonymity requires various nodes to send dummy transmissions to confuse the eavesdropper, our results show that the additional cost in terms of energy, delay and throughput can be alleviated using network coding. The key intuition is that dummy transmissions can be replaced by coded transmissions, which carry useful information. For the case of a line network with N nodes supporting coded flows, we derive closed-form expressions, which show that anonymity comes at no cost in terms of throughput if at least one of the destinations is two hops away. The average per packet delay is shown to increase by at most 50%.**

I. Introduction

With more and more communications taking place mediated by technology, security concerns are at the top of the table. In wireless networks, the broadcast nature of the wireless media allows for malicious nodes to obtain information from other nodes' transmissions. While end-to-end security mechanisms can provide some level of confidentiality, authentication, and integrity of the transmitted data [1], other valuable information, such as the active sessions among communicating entities, may be easily determined via a traffic analysis attack. Network-based anonymity techniques may allow us to hide this information from malicious nodes. Previous work in this area has focused on providing anonymity for the source (e.g. *crowds* [2]), the destination (e.g. *k-anonymity* [3]) or both (e.g. *onion-routing* [4], of which *Tor* [5] is a well-known implementation), using different mechanisms and eavesdropper models. However, if we consider a global eavesdropper, i.e., an eavesdropper that overhears all transmissions and their timings, the schemes previously mentioned have several vulnerabilities. Since in *k-anonymity* the sender of a packet is easily disclosed, the destination might be inferred based on the observation of the sender of the response. Also in the case of *crowds*, by observing the first node to transmit, a global eavesdropper may infer the true originator of a packet, specially if the number of active sessions is small. In *onion-routing*, even though each intermediate routing node does not know the original sender or the intended recipient of a message, a global eavesdropper is able to identify them, or at least segments of the route between them, by means of a classical traffic analysis attack [1]. Even though *mixing* [6] might be used to confuse a local eavesdropper, a global eavesdropper will still know which nodes are involved and may infer the active sessions in the network.

Motivated by this problem, the main contributions of this paper are: i) a novel and systematic approach to the design of fixed transmission schedules that provide perfect information-theoretic anonymity for active sessions in the network, ii) analysis in terms of delay and energy costs of anonymity as well as throughput costs, and iii) network coding techniques to reduce the cost of anonymity.

The idea of a fixed schedule was previously presented by Newman-Wolfe and Venkatraman [7], [8] and by Radosavljevic and Hajek [9]. However, references [7] and [8] focus on the case of a fully connected network and do not exploit the wireless network topology to schedule simultaneous, non-interfering transmissions across the network to increase the throughput. A different approach to anonymity in wireless networks vis-à-vis a global eavesdropper was presented by [10], where each node generates a random transmission schedule, statistically independent of the session and of the schedules of other nodes in the network. This yields a distributed approach, but guarantees no on-time packet delivery guarantee for packets with strict deadlines.

Our work proposes the use of *network coding* as a means to reduce the cost of anonymity in the network. First introduced in [11], network coding gives intermediate nodes the opportunity to operate on the information payload instead of just forwarding or replicating it. One major finding in [11] is that the maximum flow of information in a general network can not be reachable if network coding is not used. Network coding has also proved itself valuable in the case of unicast sessions on wireless networks, as explained in [12]. In particular, the use of network coding in wireless line networks has been discussed in [13] and [14]. Some previous work has already

This work was partially supported by the Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) under grant CMU-PT/CPS/0046/2008 (Vital Responder Project) and by the European Commission under grant FP7–INFSO–ICT–215252 (N-Crave Project).

related the topics of security and network coding. In [15], a low-complexity cryptographic scheme that takes advantage of random linear network is presented. An information-theoretic cryptanalysis of network coding is presented in [16]. In [17], the authors propose a scheme to achieve flow untraceability that makes use of network coding. However, the authors focus on information that may be derived from the observation of the packet coefficients and not at the timings of the transmissions.

The remainder of the paper is organized as follows. In Section II, we introduce the models, assumptions and measures. We also explain the problem by means of an example. In Section III, we provide sufficient conditions for anonymity in wireless line network. In Section IV, we present a systematic approach to construct anonymous transmission schedules. In Section V, we study the performance under our anonymous schemes. In Section VI, we analyse in detail the costs of our method for the case of a wireless line network of size N. Finally, the paper concludes with Section VII.

II. Preliminaries

We start by describing our network model, the threat model and the performance metrics under consideration. We complete the section by stating the problem and illustrating it by means of a simple, yet representative example.

A. Network Model

We represent the network by a directed hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of all nodes in the network and $\mathcal{A}$ is the set of all hyperarcs in the network. Each transmitting node $N_i \in \mathcal{N}$ is associated to one hyperarc $(i, J_i) \in \mathcal{A}$, where J_i is a nonempty subset of nodes in $\mathcal{N}$ that are in the wireless transmission range of N_i. We consider that each node transmits with a fixed power and thus has a fixed wireless transmission range, corresponding to one hyperarc (i, J_i).

B. Assumptions

We assume that the hyperlinks $(i, J_i) \in \mathcal{A}$ are lossless and therefore a transmission from node N_i will reach every node $N_j \in J_i$. The nodes have a half-duplex constraint, i.e., one node cannot simultaneously transmit and receive a message. Furthermore, if one node receives simultaneous transmissions through more than one hyperarc, these will result in a collision. The network is assumed to operate in time slots. In one slot a node can either broadcast one constant-length packet or stay idle. The width of each time slot is equal to $\frac{1}{Cap}$. In the explanation, we assume that nodes have similar capabilities and thus the maximum capacity of each hyperlink is the same and equal to Cap. The model can be easily extended to the more general case of non-homogeneous node conditions. We consider the possibility of having several multicast sessions and treat unicast and broadcast as special cases. We assume the existence of a central entity able to schedule the transmissions of different nodes with perfect synchrony. Re-scheduling only occurs if the topology changes.

C. Threat model

We consider the threat posed by an adversary who wishes to learn the active sessions by eavesdropping the transmissions of all nodes in the network. The adversary is assumed to be computationally bounded, which means he can only break a cipher if there exists a probabilistic polynomial time algorithm for that purpose. This in turn implies that the adversary cannot break the secrecy of the standard modes we assume for link encryption, e.g. randomized modes, such as CBC (Cipher-block chaining) or CTR (Counter mode) [1], which assure packet indistinguishability from random bits. It follows that the adversary is able to determine the timing of transmitted packets but cannot distinguish between packets with content (supporting active sessions) and dummy packets (aimed at adding confusion). The adversary is further assumed to be able to identify which node is transmitting at any time slot.

D. Measures

Each transmission pattern observed by an eavesdropper is associated to a subset of sessions that can be active during that pattern, thus revealing this information to the eavesdropper. We present an information-theoretic measure of the *degree of anonymity* based on the mutual information ($I(S;T)$) between the possible active sessions (S) and our schedule (T) (Eq. (1)).

$$I(S;T) = H(S) - H(S|T) \tag{1}$$

The goal of our work is to develop transmission schedules that do not compromise the entropy of the possible active sessions (S). In a perfectly anonymous schedule, we have that $I(S;T) = 0$, or equivalently (Eq. (2)):.

$$H(S|T) = H(S) = -\sum_{s \in S} p(s) \times \log p(s). \tag{2}$$

where $p(s)$ is the probability of session s to occur. For each transmission pattern which offers perfect anonymity, we define the *cost of anonymity* associated to each session S_i possible of being active in terms of throughput (Eq. (3)), delay (Eq. (4)) and energy (Eq. (5)). The cost is defined so that it is always greater or equal than one, where a cost of one corresponds to no additional cost, while a value larger than one implies a cost to providing session anonymity.

$$C_{T,S_i} = \frac{Max\ throughput\ of\ S_i}{Max\ throughput\ of\ S_i\ in\ anonymous\ pattern} \tag{3}$$

$$C_{D,S_i} = \frac{Min\ delay\ of\ S_i\ in\ anonymous\ pattern}{Min\ delay\ of\ S_i} \tag{4}$$

$$C_{E,S_i} = \frac{Min\ energy\ of\ S_i\ in\ anonymous\ pattern}{Min\ energy\ of\ S_i} \tag{5}$$

We define throughput as the number of packets that are successfully input into the network per unit time, i.e., they are delivered with a bounded delay. We define delay as the average of the number of time slots each packet will take to reach all destinations. We define energy as the number of transmissions required to deliver all packets to their intended destinations.

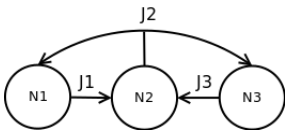

Fig. 1. Three node line network

TABLE I
POSSIBLE SESSIONS IN THREE NODE LINE NETWORK

Multicast sessions
$N_1 \to \{N_2\}$ $N_1 \to \{N_3\}$ $N_1 \to \{N_2, N_3\}$
$N_2 \to \{N_1\}$ $N_2 \to \{N_3\}$ $N_2 \to \{N_1, N_3\}$
$N_3 \to \{N_2\}$ $N_3 \to \{N_1\}$ $N_3 \to \{N_1, N_2\}$

E. Motivating example

As an example, consider the simple three node line network as depicted in Figure 1. The possible multicast sessions are listed in Table I. If we assume that all sessions are equally probable, the entropy of the possible sessions is $H(S) = \log 9$.

Imagine that only the $N_1 \to N_2$ session is active. Clearly, the schedule of transmissions that maximizes throughput and minimizes delay and energy consists of N_1 transmitting in every time slot (see Table II). However, based on the observation of this pattern, a global eavesdropper would easily conclude that only the $N_1 \to N_2$ session could be happening. The entropy of the active sessions given this pattern is $H(S|T) = 0$.

Let us now consider the transmission schedule in Table III. We only represent three instants of time, but the pattern would repeat itself through time. By observing this pattern, a global eavesdropper cannot tell which session is active. Since all sessions seem possible and equally probable to the eavesdropper, we easily see that the entropy of the active sessions given the pattern in Table III is $H(T|S) = H(S) = \log 9$. The schedule is providing perfect anonymity according to the definition in Section II-D.

However, by using this schedule of transmissions to serve the $N_1 \to N_2$ session instead of the one in Table II, we are incurring additional costs. In fact, the attainable throughput for that session is three times slower (i.e., $C_{T,N_1 \to N_2} = 3$), the energy spent is tripled (i.e., $C_{E,N_1 \to N_2} = 3$), while the packet delay remains constant (i.e., $C_{D,N_1 \to N_2} = 1$). Our goal is to provide a systematic way to develop schedules of transmissions that simultaneously maximize the degree of anonymity and minimize the cost of attaining this anonymity. Although one of our main concerns is to reduce the cost in terms of throughput, we also provide schedules that minimize delay and reduce energy consumption.

In this paper we will focus on the particular case of line networks of arbitrary size. In order to derive closed-form expressions for the cost of anonymity, we will focus on the following space of possible sessions, which we find representative:

- one general *multicast session*, which can represent the sharing of a file. We will specify this session with a source node N_i and a set of receiving nodes $\{N_{i-m}...N_{i+n}\}$. N_{i-m} refers to the node that is m hyperlinks to the left of the source node, while N_{i+n} refers to the node that is n hyperlinks to the right. Note that in the case of line networks, if a node that is k hops away receives a message, all nodes that less than k hops away will also receive it.
- *two simultaneous unicast sessions* with two nodes $N_i, N_{i+n} \in \mathcal{N}$ communicating with each other, for example a VoIP conversation.

TABLE II
(A) NON ANONYMOUS TRANSMISSION PATTERN, (B) POSSIBLE SESSIONS

	t_1	t_2	t_3
N_1	x	x	x
N_2			
N_3			

(a)

Possible sessions
$N_1 \to N_2$

(b)

TABLE III
(A) ANONYMOUS TRANSMISSION PATTERN, (B) POSSIBLE SESSIONS

	t_1	t_2	t_3
N_1	x		
N_2		x	
N_3			x

(a)

Possible sessions
$N_1 \to \{N_2\}$ $N_1 \to \{N_3\}$ $N_1 \to \{N_2, N_3\}$
$N_2 \to \{N_1\}$ $N_2 \to \{N_3\}$ $N_2 \to \{N_1, N_3\}$
$N_3 \to \{N_2\}$ $N_3 \to \{N_1\}$ $N_3 \to \{N_1, N_2\}$

(b)

III. SUFFICIENT CONDITIONS FOR ANONYMITY

We start by specifying some sufficient conditions for anonymity in wireless line networks, for the space of possible sessions considered.

A. Routing

We begin by considering the case where routing is allowed.

Proposition 3.1: Let z_i be the rate of transmission of node N_i. It is possible to serve anonymously every possible session where the source nodes are inputing new packets at rate R, if $z_i = 2R \quad \forall i \in \mathbb{N}$.

Proof: For the case of a multicast session, each packet input into the network by node N_i will be delivered to nodes $\{N_{i+1}...N_{i+n}\}$ through hyperlinks $(k, J_k), i \leq k < i+n, k \in \mathbb{N}$, which will require one transmission from each node N_k. Similarly, each packet will be delivered to node $N_{i-1}...N_{i-m}$ through hyperlinks $(k, J_k), i-m < k \leq i, k \in \mathbb{N}$, which will require one transmission from each node N_k. Therefore, all nodes envolved are only required to transmit at the same rate as the source node.

For the case of two unicast sessions between nodes N_i and N_{i+n}, each packet input into the network by node N_i will be delivered to node N_{i+n} through hyperlinks $(i, J_k), i \leq k < i+n, k \in \mathbb{N}$, which will require one transmission from each node N_k. Similarly, each packet input into the network by node N_{i+n} will be delivered to node N_i through hyperlink $(k, J_k), i < k \leq i+n, k \in \mathbb{N}$, which will require one transmission from each node N_k. Each intermediate node is required to transmit at rate R for each of the sessions, giving a total of a rate per node of $2R$. ■

B. Network coding

We now consider that nodes perform network coding.

Proposition 3.2: Using network coding, we are able to serve anonymously every possible session where the source nodes are inputing new packets at rate R, if $z_i = R \quad \forall i \in \mathbb{N}$.

Proof: The proof for the multicast session rate is the same as for Prop. 3.1. For the bidirectional unicast, we show that inter-session network coding can serve the unicast sessions at rate R. Let $Y_i(t)$ be the transmission from node N_i through hyperlink (i, J_i) at time t in a network of size N. Assume that the end nodes are exchanging a stream of packets $X_1(t)$ and $X_2(t)$ and so $Y_1(t) = X_1(t) \times \mathbf{1}_{\{t=1+3k, k\in\mathbb{Z}\}}$ and $Y_N(t) = X_2(t) \times \mathbf{1}_{\{t=1+3k, k\in\mathbb{Z}\}}$, where $\mathbf{1}_{\{f\in F\}}$ denotes the indicator function being one when $f \in F$ and zero otherwise. An intermediate node will always perform coding and transmit a combination of the last packets received from each side of the network by sending $Y_i(t) = X_1(t-(n-1)) \oplus X_2(t - 2\times(N-n)) \times \mathbf{1}_{\{t=n+3k, k\in\mathbb{Z}\}}$. This technique is refered to as reverse car pooling in the network coding literature [12]. Each of the neighbours will gain a packet, recovered from the one they already knew and from the combination received. This concludes the proof. ■

IV. Transmission Schedules

In this section, we propose an approach to construct the transmission schedules that guarantee anonymity, by respecting the sufficient conditions proposed in Section III, and that reduce the cost in throughput of the active sessions.

A. Conflict graph

As proposed in [18], maximizing the simultaneous transmissions by different nodes' in the network will maximize the achievable throughput. For that, we make use of an auxiliary undirected graph $\mathcal{C} = (\mathcal{V}, \mathcal{A})$, called *conflict graph*, similar to what was used in [18] or [19]. In our formulation, the vertices in the conflict graph correspond to nodes in the network, that is $\mathcal{V} = \mathcal{N}$. Any pair of vertices will be connected if an interference may arise from a simultaneous transmission of both nodes. According to the interference model detailed in Section II-B, two vertices V_i and V_j will be connected if $N_j \in J_i$ and $N_i \in J_j$, and also if there is one node N_k such that $N_k \in J_i$ and $N_k \in J_j$.

B. Stable sets

It is straightforward to conclude that two nodes N_i and N_j may transmit simultaneously if their corresponding vertices V_i and V_j are not connected in $\mathcal{C}$, that is, $A_{ij} \notin \mathcal{C}$. More generally, a given set of nodes $N_i \in \mathcal{N}$ may transmit simultaneously if no two of them are adjacent in $\mathcal{C}$. In other words, for every possible pair of nodes in this set, there is no edge connecting them. We call such set a *stable set* or *independent set*. The problem of having as many simultaneous transmissions as possible is equivalent to finding independent sets containing as many vertices as possible. In this context, a *maximal independent set* is an independent set which cannot be further grown, i.e., if any other vertex is added to the set it will force it to have a connection. A given graph may have several maximal independent sets, the largest of which is called the *maximum independent set*. Finding a maximal stable set is an easy problem and may be solved in polynomial time using a simple greedy algorithm, such as the one in [20], finding the maximum independent set of a graph is well known to be a NP hard problem, except for some types of graphs.

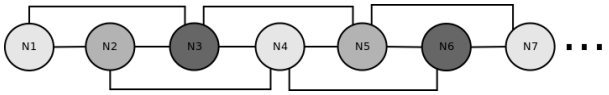

Fig. 2. Coloring of line network conflict graph.

C. Graph coloring

Rather than aiming at having the largest number of nodes transmit in each time instant, which is a classical scheduling goal, our problem in anonymity is to have *every node* transmit in the smallest amount of time possible. In other words, we want to find the minimum number of stable sets in the conflict graph which, together, span the whole graph. This corresponds to a problem known as *graph coloring*. In short, we aim to color the vertices of a graph in such a way that no two adjacent vertices share the same color. This way, each set of vertices equally colored must form an independent set. The minimum number of colors needed to color a graph $\mathcal{G}$ is called its *chromatic number* and is denoted by $\mathcal{X}(\mathcal{G})$. In the context of our problem, all the vertices equally colored in $\mathcal{C}$ represent nodes that will transmit simultaneously. On the other hand, the chromatic number of the conflict graph, $\mathcal{X}(\mathcal{C})$, represents minimum amount of time it will take for every node to transmit and will be directly related to the achievable throughput under anonymity conditions. Unfortunately, graph coloring is also well known to be a NP hard problem. For a general topology case, we can use heuristics, such as the one in [21], which only achieve suboptimal solutions but are more computationally friendly. In the case of the line network, determining the chromatic number is much simpler.

Theorem 4.1: It is possible to color the conflict graph of an arbitrary size line network using only three colors.

Proof: It is not possible to color the graph using only two colors, because N_1, N_2 and N_3 are all adjacent nodes in $\mathcal{C}$ and need to have different colors. A solution with $\mathcal{X}(\mathcal{C}) = 3$ is shown in Figure 2, resulting in the pattern shown in Table IV. ■

V. Throughput, energy and delay performance evaluation

In this section, we analyse the performance within our anonymous schedule focusing on different possibilities for the transmission schedule. However, the reader should keep in mind that in the end only one of the schedules should be used and should remain fixed independently of the active sessions.

1) Routing: Let us first consider the case of routing with a single multicast session. We define one period of the transmission pattern as the interval of time (measured in time slots) each source node will input a new packet into the network.

By Proposition 3.1, every node should transmit at the same rate as the source. While using the anonymous pattern in

TABLE IV
N NODE LINE NETWORK ANONYMOUS TRANSMISSION PATTERN

	t_1	t_2	t_3	t_4	t_5	t_6
N_1	x			x		
N_2		x			x	
N_3			x			x
N_4	x			x		
N_5		x			x	
N_6			x			x
.	.	.	.	.	.	.
.	.	.	.	.	.	.
N_{1+6k}	x			x		
N_{2+6k}		x			x	
N_{3+6k}			x			x
N_{4+6k}	x			x		
N_{5+6k}		x			x	
N_{6+6k}			x			x

TABLE V
PERFORMANCE WITH ANONYMITY IN LINE NETWORK OF SIZE N

	Unicast $N_i \leftrightarrow N_{i+n}$		Multicast $N_i \rightarrow \{N_{i-m}...N_{i+n}\}$
n	1	≥ 2	$\forall$
m	n.a.		$\forall$
Throughput	$\frac{2\times Cap}{3}$	$\frac{Cap}{3}$	$\frac{Cap}{3}$
Energy	N	$2N$	N
Delay	$2n-1$		$max(n, 2m-1)$ $max(2n-1, m)$

Table IV, the source is able to input a new packet every 3 time slots and every node will also transmit once in this period. The maximum throughput attainable is then equal to $\frac{Cap}{3}$ and the energy spent is equal to the number of nodes in the network, N. The throughput is *not* dependent on the size of the network.

Depending on the order in which the different colors transmit, the packets will experience a different rightwards and leftwards delay. We know that for the rightwards flow of information we need nodes N_i, N_{i+1}, ..., N_{i+n-1} to transmit in this order. This corresponds to having colors C_1, C_2, C_3 transmit in this order. Similarly, a leftwards flow of information will require colors C_1, C_3, C_2 to transmit in this order. Since the pattern will repeat itself indefinitely, there is a circular symmetry. In other words, having the order C_1, C_2, C_3 is exactly the same as having C_2, C_3, C_1. Therefore, in this case there are only two possible for the orders of the transmissions, which are C_1, C_2, C_3 (shown in Table IV) and C_1, C_3, C_2. The different packet delays obtained are summarized in Table V, along with the performance in throughput and energy.

We now consider the possibility of having two simultaneous unicast sessions. By Proposition 3.1, every node should transmit at twice the rate the sources are inputing information. Therefore, for each packet input into the network every node needs to transmit two times. By observing the anonymous pattern in Table IV, we see that it takes 6 time slots for every node to transmit twice and thus each source is only able to input a new packet every 6 time slots. The maximum throughput attainable is then equal to $2 \times \frac{Cap}{6} = \frac{Cap}{3}$ and the energy spent to deliver both packets is equal to $2N$. In the particular case of a unicast session between two neighbour nodes, since there are no intermediate nodes, the transmission pattern only needs to have a period of 3. This allows us to achieve a higher throughput and to expend less energy in this case. Once more, the order in which the different colors transmit will determine the packet delays. However, we now have more freedom to distribute the different colors through time, since the pattern only needs to have a period of 6. We need to allocate two instances of each of the three colors to six time slots. Again considering the circular symmetry, we end up with 30 different possible permutations. One way of proving this is to fix the first color to be C_1, in order to discard the circular symmetry. We then have five positions available for the other C_1 and four positions for the other two C_2s. The C_3s will be in the remaining spots. We then have $5 \times \binom{4}{2} = 30$ possibilities. In each pattern, one of the occurances of C_1 will be used for the rightwards flow of information and the other for the leftwards one. The same applies to C_2 and C_3. For each pattern generated, there are in $2^3 = 8$ different ways of using the transmissions, which also result in different delays for the rightwards and leftwards flow of information.

Let us consider the case of having one of the 30 possible patterns, formed by $C_1, C_3, C_2, C_2, C_3, C_1$ in this order. Consider also one of the eight possible ways of using the transmissions, where r means a transmission is used for the rightwards flow and l means it is used for a leftwards flow. For example, $C_1(r), C_3(l), C_2(r), C_2(l), C_3(r), C_1(l)$. We see that we can run through the sequence $C_1(r), C_2(r), C_3(r), C_1(r)$ with a delay of 2 between each valid transmission. Also, we can run through the sequence $C_1(l), C_3(l), C_2(l), C_1(l)$ with a delay of 2 between each valid transmission. We define a per edge delay as the total delay a packet experienced to get from the source to the destination, divided by the number of hyperlinks it traversed. In this case, we have an average per edge delay of two and therefore to operate in point A in Figure 3, which is the best we can achieve.

If we take into account all the possibilities and perform a similar analysis as we did for the above example, we will have four operating points in terms of delay (A, B, C and D), as shown in Figure 3. It is also possible to create a pattern that is a convex combination of the 4 operating points, residing inside the square in Figure 3. All the performance information is summarized in Table V. Clearly, configurations yielding point A provide the best delay performance.

2) Network coding: As explained in Proposition 3.2, we can attain perfect anonymity with every node transmitting at the same rate as the sources. Since it is possible to make every node transmit once in 3 time slots, with network coding we can serve every possible session with a period of 3, thus enhancing the sessions' throughput without compromising anonymity. Furthermore, since each transmission forwards at the same time one packet rightwards and another leftwards, this will allow for savings in delay and in the number of transmissions. Comparatively to the case of no network coding, the throughput is doubled and the transmissions are reduced by half, in the case of unicast sessions. In the case of multicast sessions the performance will not change.

Again, the order in which the transmission possibilities

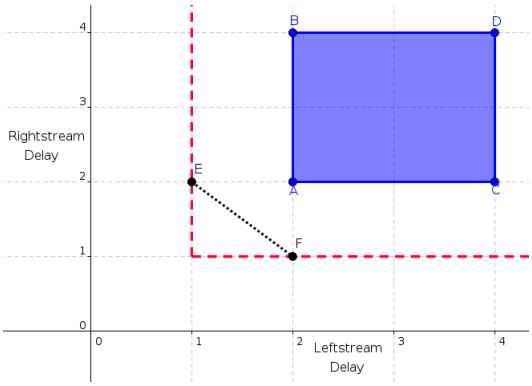

Fig. 3. Per edge average delay operating points in anonymous transmission pattern using i) routing (solid) ii) network coding (dotted). The lower bounds for delay are represented by the dashed line.

TABLE VI
PERFORMANCE OF UNICAST SESSION BETWEEN N_i AND N_{i+n} USING NETWORK CODING WITH ANONYMITY IN LINE NETWORK OF SIZE N

n	Throughput	# Transmissions	Average Packet Delay
$\forall$	$\frac{Cap\times 2}{3}$	N	$\frac{3n-1}{2}$

appear will determine the packet delays. The two points of operation (E and F) are shown in Figure 3 in terms of average delay per edge. Again, we may also operate in any point belonging to a convex combination of these points, by alternating between these two patterns. All the operating points allow a better performance than in the no network coding case.

VI. Cost of Anonymity

In this section, we present closed-form expressions for the cost of anonymity in a N node line network. The measures defined in Section II-D will be considered. All the costs were obtained by deriving a best performance bound and comparing it with the one under anonymity conditions in Section V.

3) Routing: We start by considering the case where only routing is allowed.

Proposition 6.1: The cost of hiding a multicast session between N_i and the set of nodes $N_{i-m}...N_{i+n}$ in a N node line network is

$$C_{T,N_i\rightarrow\{N_{i-m}...N_{i+n}\}} = \begin{cases} 3 & max(n,m)=1 \\ \frac{3}{2} & max(n,m)=2 \\ 1 & max(n,m)\geq 3 \end{cases}$$

$$C_{E,N_i\rightarrow\{N_{i-m}...N_{i+n}\}} = \frac{N}{n+m-1} \quad (6)$$

$$C_{D,N_i\rightarrow\{N_{i-m}...N_{i+n}\}} = \frac{max(n,2m-1)}{max(n,m)}$$

Proof: In the case of throughput, when the destinations are neighbours of the source node, this one is able to transmit at maximum capacity without anonymity concerns. If the farthest destination is two edges apart, the source node will only be able to input a new packet every two time slots, since the time slot immediately after is reserved for the intermediate node to route the packet to the destination. By a similar reasoning, each source node can only input a new packet every three instants of time in the case the farthest destination is three or more links away. The minimum number of transmissions possible is $n+m-1$, one for the inicial transmission from node N_i, $n-1$ transmissions send the information to the leftmost destination and $m-1$ to send them to the rightmost one. Since the farthest destination is $max(n,m)$ links apart, the minimum possible delay is $max(n,m)$. Depending on the anonymous pattern used, a different cost from the above specified could be obtained for the case of delay. ■

TABLE VII
MAXIMAL THROUGHPUT TRANSMISSION PATTERN FOR $A_1 \leftrightarrow A_5$ UNICAST SESSION ($n=4$)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
N_1		x				x				
N_2	x		x		x		x			
N_3				x	x				x	x
N_4		x				x		x		x
N_5	x						x			

Proposition 6.2: The cost of hiding two unicast sessions between nodes N_i and N_{i+n} in a N node line network is

$$C_{T,N_i\leftrightarrow N_{i+n}} = \begin{cases} \frac{3}{2} & n=1,2,3 \\ \frac{6}{5} & n=4 \\ 1 & n\geq 5 \end{cases} \quad (7)$$

$$C_{E,N_i\leftrightarrow N_{i+n}} = \begin{cases} \frac{N}{2n} & n=1 \\ \frac{N}{n} & n\geq 2 \end{cases}, \quad C_{D,N_i\leftrightarrow N_{i+n}} = 2-\frac{1}{n}$$

Proof: We start by deriving the upper bounds for throughput. The case of $n=1$ is trivial, since nodes will alternately send their packets. In the case of $n=2$, the throughput is reduce to half, since we now have one intermediate node which needs to forward both transmissions from N_1 and N_3, which with routing requires two additional time slots. It is possible to serve the case of $n=3$ without a reduction in throughput relatively to the $n=2$ case, by making use of a simultaneous transmission of N_1 and N_4 and another one by N_2 and N_3. This last transmission will result in collision, since node N_2 will not be able to hear the information from node N_3 and vice-versa. However, since the information is actually destinated to nodes N_1 and N_4, this collision is harmless. We now look at the case of $n=4$. It is possible for each source node to inject a new packet every 5 instants of time, by alternating between the patterns shown in Table VII, the first between $t=1$ and $t=5$, and the second between $t=6$ and $t=10$. We again schedule simultaneous transmissions which will result in harmless collisions. We see that each intermediate node will, on the end of two periods, have transmitted twice the times of the source nodes.

We now look at the cases of energy and delay. Since each source node inputs one packet per period and each intermediate node transmits twice to account for both flows, this gives a minimimum number of transmissions of $T = 1+1+2\times(n-1) = 2n$. The lowest possible average delay in a unicast session between nodes N_i and N_{i+n} is n, since the nodes are n hops apart. ■

We can achieve anonymity with no cost in throughput as long as one destination is at least five hops. The cost in energy is high if we are hiding sessions between nodes that are a few links away, but quickly drops as this distance increases. For any case, the packet delay is at most doubled.

4) Network coding: In the case of line networks, network coding brings no benefit in the case where only one multicast session is active. As detailed in Proposition 3.2, the benefit comes when we consider two simultaneous unicast sessions.

Proposition 6.3: The cost of hiding two unicast sessions between nodes N_i and N_{i+n} in a N node line network using network coding is

$$C_{T,N_i \leftrightarrow N_{i+n}} = \begin{cases} \frac{3}{2} & n = 1 \\ 1 & n \geq 2 \end{cases} \tag{8}$$

$$C_{E,N_i \leftrightarrow N_{i+n}} = \frac{N}{n+1}, \qquad C_{D,N_i \leftrightarrow N_{i+n}} = \frac{3}{2} - \frac{1}{2n}$$

Proof: We first need to compute the higher bounds for throughput. The case of $n = 1$ is trivial, since each time slot one of the nodes will input its packet. If we consider the case of $n = 2$, at least two instants of time are needed for each of the sources to input their packet, and another one for the intermediate node to perform coding, giving a throughput of $\frac{2}{3}$. As summarized in Table VI, this is exactly the throughput we achieve by using our pattern for any case of $n \geq 2$, which means that anonymity is not compromising throughput in any of these cases. Using network coding, both sources and intermediate nodes only need to perform one transmission per period, giving a total of $n + 1$ transmissions. The lowest possible average delay in a unicast session between nodes N_i and N_{i+n} is n, since the nodes are n hops apart. ■

With network coding, there is no cost in throughput as long as one destination is at least two hops away. The cost in energy is still high if we are hiding sessions between nodes that are a few links away, but also drops as this distance increases. Packet delay is increased by at most a factor of 1.5.

VII. Conclusion and future work

We provide a systematic way to build transmission patterns which provide perfect information-theoretic anonymity to the active sessions of a the line network. Closed-form expressions for the incurred costs in terms of throughput, energy and delay were also obtained. When no network coding is used, we proved that anonymity comes with no additional cost in throughput if at least one destination is five links away and that the delay is at most doubled. We demonstrated benefits in performance of using network coding, namely that the same anonymity is attainable with double throughput and half the energy. We also showed that the cost in throughput is zero except for sessions between neighbour nodes and that the delay is increased by at most one half, when using network coding.

In the future, we expect to study trade-offs between anonymity and performance as well as to extend the analysis for a general network case. We have already hinted how to design schedules for a general network topology using graph coloring. Furthermore, network coding may have a significant impact when we consider lossy networks, because network-coded packets can be sent instead of dummy packets to both maintain the transmission schedule and increase redundancy.

References

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Boston, USA: Pearson Education, 2010.

[2] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, pp. 66–92, November 1998.

[3] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, October 2002.

[4] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Proc. of the First Int. Workshop on Inf. Hiding*. London, UK: Springer-Verlag, May 1996, pp. 137–150.

[5] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proc. of the 13th conf. on USENIX Security Symposium - Volume 13*, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, August 2004, p. 21.

[6] J. Ghaderi and R. Srikant, "Towards a theory of anonymous networking," *Computing Research Repository*, vol. abs/0908.1805, August 2009.

[7] R. Newman-Wolfe and B. Venkatraman, "High level prevention of traffic analysis," in *Computer Security Applications Conf., 1991. Proc., Seventh Annual*, San Antonio, Texas, USA, December 1991, pp. 102 –109.

[8] B. Venkatraman and R. Newman-Wolfe, "Transmission schedules to prevent traffic analysis," in *Computer Security App. Conf., 1993. Proc., Ninth Annual*, Orlando, Florida, USA, December 1993, pp. 108 –115.

[9] B. Radosavljevic and B. Hajek, "Hiding traffic flow in communication networks," in *Military Communications Conf., 1992. MILCOM '92, Conference Record. 'Communications - Fusing Command, Control and Intelligence'., IEEE*, San Diego, California, USA, October 1992, pp. 1096 –1100 vol.3.

[10] P. Venkitasubramaniam, T. He, and L. Tong, "Anonymous networking amidst eavesdroppers," *Information Theory, IEEE Trans. on*, vol. 54, no. 6, pp. 2770 –2784, June 2008.

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

[12] S.-Y. K. Yunnan Wu, Philip A. Chou, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Conf. on Inf. Scie. and Syst.*, Baltimore, USA, March 2005.

[13] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," in *Information Theory, 2005. ISIT 2005. Proc. International Symposium on*, Adelaide, Australia, September 2005, pp. 1853 –1857.

[14] U. Niesen, C. Fragouli, and D. Tuninetti, "On capacity of line networks," *Information Theory, IEEE Trans. on*, vol. 53, no. 11, pp. 4039 –4058, November 2007.

[15] J. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," in *Communications, 2008. ICC '08. IEEE International Conference on*, Beijing, China, May 2008, pp. 1750 –1754.

[16] L. Lima, J. Vilela, J. Barros, and M. Medard, "An information-theoretic cryptanalysis of network coding - is protecting the code enough?" in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, Auckland, New Zealand, December 2008, pp. 1 –6.

[17] J. Wang, J. Wang, C. Wu, K. Lu, and N. Gu, "Anonymous communication with network coding against traffic analysis attack," in *INFOCOM, 2011 Proc. IEEE*, Shangai, China, april 2011, pp. 1008 –1016.

[18] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 66–80.

[19] D. Traskov, M. Heindlmaier, M. Medard, R. Koetter, and D. Lun, "Scheduling for network coded multicast: A conflict graph formulation," in *GLOBECOM Workshops, 2008 IEEE*, New Orleans, Louisiana, December 2008, pp. 1 –5.

[20] M. Luby, "A simple parallel algorithm for the maximal independent set problem," in *Proc. of the seventeenth annual ACM symposium on Theory of computing*, ser. STOC '85. New York, NY, USA: ACM, 1985, pp. 1–10.

[21] D. S. Hochbaum, "Efficient bounds for the stable set, vertex cover and set packing problems," *Discrete Applied Mathematics*, vol. 6, no. 3, pp. 243 – 254, 1983.