Secure Network Coding with a Cost Criterion

Jianlong Tan, Muriel Médard
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
E-mail: {jianlong, medard}@mit.edu

Abstract — **Network cost and network security are two of many network parameters that are important to network users. While these two parameters have been separately considered in coded networks (networks that employ network coding), a joint investigation of them both has not been done yet to our knowledge, thus providing the motivation for this work. In this paper, we consider the situation where a set of messages is to be multicasted across the network, and a known subset of these messages is of interest to a wiretapping adversary. The problem that we attempt to solve is to find a network coding scheme that has both a low network cost and a low probability of the wiretapper being able to retrieve all the messages of interest. We make use of random linear codes in anticipation for decentralized implementation of the scheme, and focus on the problem of finding the multicast subgraph. As an exact algorithmic solution is difficult, we propose two heuristic solutions, and compare their performances to traditional routing through a simulation study. Our results suggest that network coding can be more effective than routing for this low cost and secure data multicast problem, especially when the links are not easily tapped.**

1. INTRODUCTION

For any communication network or protocol, there are many performance parameters that are of importance to the network users, such as throughput rates and network robustness. When the concept of network coding was introduced by Ahlswede et. al. [1], the authors already demonstrated that network coding can achieve higher throughput rates than traditional routing.

One important performance parameter is the network cost incurred for a given set of connections, and the complexity associated with the computation of the subgraphs needed to provide the connections. While the minimum cost multicast problem in routed networks requires the finding of a directed Steiner tree, which is NP-hard, the same problem in coded networks can be solved by a linear program in polynomial time [2], and also be implemented in a decentralized manner [3]. In addition, simulation results have shown that network coding can provide the multicast connections at a lower cost than traditional routing [3], [4].

Another important performance parameter is the security of the network. Cai and Yeung [5] considered the problem of using network coding to achieve perfect information security against a wiretapper who can eavesdrop on a limited number of network links, and presented the construction of a secure linear network code for this purpose. Feldman et. al. [6] generalized and simplified the solution by showing that the problem is equivalent to finding a linear code with certain generalized distance properties. In a different setting, Ho et. al. [7] showed that randomized network coding is useful in detecting Byzantine modification of data packets, thus providing data security against Byzantine attackers who arbitrarily modify data packets in the network.

While both network cost and security have been separately investigated in the network coding literature, they have not been jointly investigated yet to our knowledge, thus providing the motivation for this paper. Here, we consider the situation where a set of messages is to be multicasted across the network, of which a subset is of interest to a wiretapper. The problem that we want to solve is to multicast the messages at a low cost, while keeping the network vulnerability — defined as the probability that the wiretapper is able to retrieve all the messages of interest — low. While network security is not limited to the resilience of the network against wiretapping, the other notions of security are beyond the scope of this paper.

In general, we expect a trade-off between network cost and network vulnerability. For instance, in routed networks, a cheapest cost approach to a unicast connection usually selects a single path. However, when the connection is to be resilient against wiretapping, multiple disjoint paths may be used, which may increase the network cost [8], [9], [10], [11].

To illustrate further this trade-off and to show that network coding has the potential of achieving a lower network vulnerability than traditional routing, consider the network shown in Figure 1(a) where each network link has unit capacity and unit cost. Two random processes (denoted X_1 and X_2) are to be multicasted from the source nodes s_1 and s_2 to the sink nodes t_1 and t_2, against a wiretapper who is interested in obtaining the random process X_1. The probability that any one particular link is tapped is 0.01, and edges are assumed to be tapped independently of one another.

Figures 1(b) to 1(e) show four different methods of achieving the multicast, and Table 1 shows the corresponding network costs and vulnerabilities. In Figures 1(b) and 1(c), each process is transmitted at unit rate, while in Figures 1(d) and 1(e), each process X_i is transmitted at a rate of two by splitting it into two processes X_{i1} and X_{i2}. Figure 1(b) shows the single path routing solution that minimizes the network cost, while Figure 1(c) shows the non-trivial single path network coding solution (note that the trivial solution is identical to the routing solution). Figure 1(d) shows the multipath routing solution and Figure 1(e) shows the multipath network coding solution.

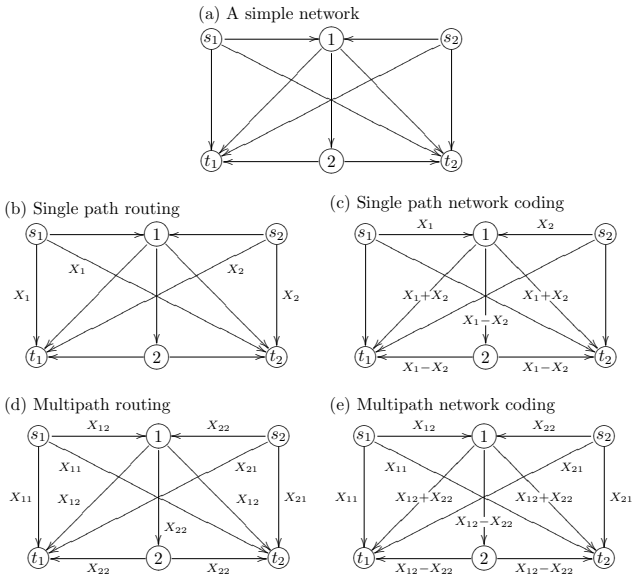

Figure 1: A simple network example.

Table 1: Network costs and vulnerabilities for the simple network.

	Average cost (per bit)	Network Vulnerability
Single path routing	2	0.020
Single path coding	3.5	0.010
Multipath routing	2.75	5.9×10^{-4}
Multipath coding	2.75	2.1×10^{-5}

From this simple example, we see that, while the single path routing solution offers the lowest network cost, it results in the highest network vulnerability. While the network vulnerability can be reduced by employing network coding or multipath routing, the network cost inevitably increases. It should also be noted that the multipath network coding solution returns the lowest network vulnerability, at a cost equal to that of multipath routing.

The rest of this paper is structured as follows. Section 2 describes the problem in greater detail. Section 3 describes the general solution to the problem, which is difficult in general, and suggests two heuristic methods that can be used to solve the problem. Simulation results using the two proposed heuristics, together with a discussion of the results, are presented in Section 4.

2. Problem formulation

A communication network is represented by a directed graph $G = (V, E)$, where V is the set of vertices (or nodes) and E is the set of edges (or links). To each edge $l \in E$, we associate two non-negative numbers c_l and d_l, which are the cost per unit flow and the link capacity, respectively. For simplicity, the edges are assumed to be free of delays, but the results still apply for networks with delays [13].

In this network, a set of r discrete independent random processes $\mathscr{W} = \{W_1, \ldots, W_r\}$ is to be transmitted to a fixed set of sink nodes, $T = \{t_1, \ldots, t_{|T|}\} \subset V$. Each random process W_i is generated at the source node s_{W_i}, and is assumed to have a constant integer entropy rate of ρ. Thus, it can be decomposed into ρ independent random processes, each with unit entropy rate. We denote the j-th such component of W_i as the random process $X_{\rho(i-1)+j}$, and we denote the source node of process X_i as s_{X_i}.

In the network, there exists an adversary who is interested in knowing a given subset of $\mathscr{W}$, denoted by $\mathscr{W}_{interest}$. Without loss of generality, we assume that the adversary is interested in the first k random processes (i.e. $\mathscr{W}_{interest} = \{W_1, \ldots, W_k\},\ k \leq r$). In terms of the unit entropy processes, we say that the adversary is interested in the set $\mathscr{X}_{interest} = \{X_1, \ldots, X_{k\rho}\}$.

This adversary is able to tap network links and retrieve all the messages that are being transmitted along them. To model this wiretapping behavior, we associate with each edge l, a number $p_l \in [0, 1]$ to denote the probability that the wiretapper will be able to retrieve the messages sent along it. Edges are assumed to be tapped independently of one another, and we let $\mathscr{Y}_{tapped}$ be the (random) set of messages transmitted along the tapped links. We define the network vulnerability ν to be the probability that the conditional entropy of $\mathscr{W}_{interest}$, given $\mathscr{Y}_{tapped}$, is zero:

$$\nu \quad := \quad P\left[H\left(\mathscr{W}_{interest} \mid \mathscr{Y}_{tapped}\right) = 0\right].$$

This definition of the network vulnerability is logical because in order for the wiretapper to figure out the identities of all the messages in $\mathscr{W}_{interest}$, he will need to make a guess out of $2^{H(\mathscr{W}_{interest} \mid \mathscr{Y}_{tapped})}$ possibilities.

Assuming that the multicast problem is feasible given the network topology, the problem to be solved here is the design of a network coding scheme that has both a low overall network cost and a low network vulnerability. To define the problem more concretely, we denote the overall network cost by the variable $\mu = \sum_{\{l \in E\}} c_l z_l$, where z_l is the amount of flow transmitted on the link l. The problem is then to minimize some function $\mathscr{F}(\mu, \nu)$, which is an increasing function of both μ and ν. Once again, as we expect a trade-off between network cost and vulnerability, it is often inadequate to minimize only μ or ν.

3. Problem solution

We separate this network coding problem into two parts. The first part deals with the finding of the coding subgraph, including the amount of information flow to be put onto each network link. The second part involves the actual network code to be implemented in the subgraph. The network code describes how data packets in the network interact with one another, and is important to ensure that the original messages can be decoded at the sink nodes.

In anticipation of a distributed implementation of the solution, we make use of random linear network codes, which are sufficient for multicast [12]. The random linear coding model

that we invoke henceforth follows that described in the original paper by Ho et. al. [13]. In particular, we let the coding subgraph be represented by $G' = (V, E')$, where every edge $l' \in E'$ has unit capacity and carries the random process $Y(l')$. The finite field used is denoted by $\mathbb{F}_{2^u}$, and the set of all transmitted processes is denoted by $\mathscr{Y}$:

$$\mathscr{Y} = \bigcup_{\{l' \in E'\}} \{Y(l')\}.$$

Since all the random processes in $\mathscr{W}$ have to be multicasted to T, we translate the problem into a single multicast problem by introducing into G, a pseudo-source node α that generates all the processes in $\mathscr{W}$ and transmits them to the actual source nodes $s_{W_1}, \ldots, s_{W_r}$ via pseudo-edges (α, s_{W_i}).

Noting that the problem of finding the minimum cost subgraph for a single multicast can be cast into a linear programming problem [2], we structure the subgraph finding problem as the following constrained optimization:

$$\begin{aligned}
&\text{minimize} \quad \mathscr{F}(\mu, \nu) \\
&\text{subject to} \quad z_l = \rho, && \forall\, l \text{ s.t. } \text{tail}(l) = \alpha, \\
&\qquad z_l \geq x_l^{(t)}, && \forall\, l \in E,\ t \in T, \\
&\qquad \sum_{\{l:\ \text{tail}(l)=v\}} x_l^{(t)} - \sum_{\{l:\ \text{head}(l)=v\}} x_l^{(t)} = \sigma_v^{(t)}, && \\
&\qquad && \forall\, v \in V,\ t \in T, \\
&\qquad d_l \geq x_l^{(t)} \geq 0, && \forall\, l \in E,\ t \in T,
\end{aligned} \tag{1}$$

where:

$$\sigma_v^{(t)} \;=\; \begin{cases} r\rho & \text{if } v = \alpha, \\ -r\rho & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases}$$

One difficulty of this optimization is that, while it is easy to compute μ given $x_l^{(t)}$ and z_l, the same is not true for ν, as it depends on the actual processes sent along the links (i.e. the network code). As an example, consider the network shown in Figure 2, with $\rho = 1$, $\mathscr{W} = \{W_1, W_2\} = \{X_1, X_2\}$, $\mathscr{W}_{interest} = \{W_1\} = \{X_1\}$. Figure 2(a) shows the values of $x_l^{(t)}$ and z_l, while Figure 2(b) shows the network code.

Consider the links $l_1 = (s_1, 1)$ and $l_2 = (2, t_2)$, which have the same $x_l^{(t)}$ and z_l values, but are carrying different processes. Note that while the tapping of l_1 will enable the retrieval of X_1, the sole tapping of l_2 gives no information about X_1. From this example, we see that ν depends not just on $x_l^{(t)}$ and z_l, but also on the actual network code.

Another difficulty is that ν and $\mathscr{F}(\mu, \nu)$ may not be convex functions of $x_l^{(t)}$ and z_l. For instance, consider the network shown in Figure 1(e), where we now assume that every edge has a capacity of two. We make a single change to the network code, whereby for the edge $(s_2, 1)$, in addition to transmitting X_{21}, we also transmit X_{22} in an uncoded manner. As the knowledge of X_{22} does not help the wiretapper in knowing the identity of either X_{11} or X_{12}, this increase in z_l for the edge $(s_2, 1)$ does not affect ν.

As a result of these difficulties, we shall look at some heuristic methods to solve the problem.

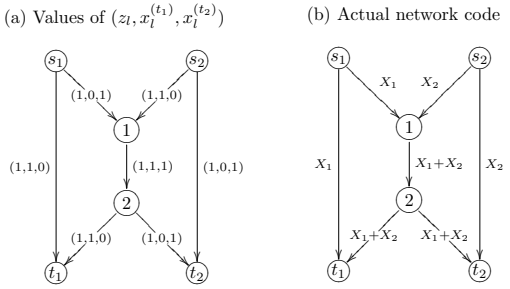

Figure 2: Network vulnerability depends on the network code.

3.1 First Heuristic

Consider the scenario where the following statements are true. While these statements are not true in general, they can be reasonable approximations to the actual network code when the processes are well-mixed throughout the network.

First, we have:

$$Y(l'_1) = Y(l'_2) \quad \Leftrightarrow \quad l'_1 = l'_2, \qquad \forall\, l'_1, l'_2 \in E'.$$

Hence, the tapping of any n links in E' by the wiretapper will provide him with a set of $\min(r\rho, n)$ linearly independent equations in terms of the input processes X_i.

Second, we have:

$$Y(l') = \sum_{i=1}^{r\rho} \zeta_{l',i} X_i, \qquad \forall\, l' \in E',$$

where:

$$\zeta_{l',i} \in \mathbb{F}_{2^u} \backslash \{0\}, \qquad \forall\, l' \in E',\ 1 \leq i \leq r\rho.$$

Hence, in order to retrieve all the messages in $\mathscr{W}_{interest}$, the wiretapper will need to have a set of exactly $r\rho$ linearly independent equations in terms of the X_i's [14].

Under these conditions, ν is then equal to the probability that the wiretapper manages to tap at least $r\rho$ links in E', and this can be obtained from the values of z_l and $x_l^{(t)}$ in the following manner.

Consider a link $l \in E$ that is carrying a flow amount $z_l \in \mathbb{R}_0^+$. Since each link in E' has unit capacity, the number of unit capacity links in E' that correspond to the link l is approximately $[z_l]$ — the nearest integer to z_l. With this, we let L_l be the discrete random variable denoting the amount of information flow successfully tapped on link l. As p_l is the probability of tapping link l, we then have the z-transform of L_l as (to avoid confusion, the transform variable z is replaced by w):

$$L_l(w) \;=\; E[w^{L_l}] \;=\; p_l w^{[z_l]} + (1 - p_l).$$

Now, let L_{total} be the random variable denoting the total amount of information flow successfully tapped in the network. Since edges are assumed to be tapped independently of one another, we have:

$$L_{total}(w) \;=\; \prod_{\{l \in E\}} L_l(w) \;=\; \sum_i P(L_{total} = i) w^i$$

and the network vulnerability is given by:

$$\nu' \;=\; P(L_{total} \geq r\rho) \;=\; 1 - \sum_{i=0}^{r\rho-1} P(L_{total} = i),$$

which is an increasing function with respect to each z_l, but may not be convex in nature.

Algorithm 1 below summarizes how to obtain ν' from the vector $z = (z_l)$. In the algorithm, k_l is the vector of polynomial coefficients of $L_l(w)$ and k_{total} is the vector of polynomial coefficients of $L_{total}(w)$, which is obtained by convolving all the k_l vectors. Note that the symbol $*$ is used to denote the convolution operation.

Algorithm 1: Given z, calculate ν'.

$z'_l \leftarrow [z_l] \quad \forall\, l \in E$;
for each $l \in E$ **do**
  $k_l \leftarrow$ zero vector of length $(z'_l + 1)$;
  if $z'_l = 0$ **then**
    $k_l[1] \leftarrow 1$;
  else
    $k_l[1] \leftarrow p_l$;
    $k_l[z'_l + 1] \leftarrow 1 - p_l$;
  end
end
$k_{total} \leftarrow k_{l_1} * k_{l_2} * k_{l_3} * \ldots$;
$h \leftarrow length(k_{total})$;
$\nu' \leftarrow 1 - \sum_{j=h-r\rho+1}^{h} k_{total}[j]$;

With this, we replace the objective function in (1) by the function $\mathscr{F}(\mu, \nu')$, which is a function in terms of z_l only. For our simulations, we considered a specific form of the objective function, given by:

$$\mathscr{F}(\mu, \nu') \quad = \quad \mu + \omega\nu',$$

where ω is a non-negative weighting variable that represents the relative importance of the network vulnerability with respect to the overall network cost.

3.2 Second Heuristic

One major drawback of the first heuristic is that the calculation of ν' can be computationally intensive, as it requires the calculation of the polynomial coefficients of $L_{total}(w)$. Because of this complexity, a different heuristic is proposed here.

As mentioned in the introduction, we first note that in routed networks, the use of disjoint data paths can decrease the network vulnerability. Returning to our problem, consider the spreading of information flow across the network links. If all the flows are concentrated along a single multicast tree, then $P(L_{total} \geq r\rho)$ can be quite high, as each used link carries a large amount of flow, and one only needs to tap a few of them to have $L_{total} \geq r\rho$. However, if the flows are spread out across network links instead, one will need to tap more network links to have $L_{total} \geq r\rho$. Hence, we expect the network vulnerability to decrease as information flow is spread across the network.

A straightforward method to spread information flow across network links is to introduce to each link $l \in E$, a strictly convex and increasing cost function in terms of z_l. For our simulations, we considered the addition of a quadratic cost function to each link. Specifically, the objective function in (1) is replaced by the following function:

$$\mu + \omega'\mu_{quad},$$

where

$$\mu_{quad} \quad = \quad \sum_{\{l \in E\}} p_l \left(z_l/\rho\right)^2,$$

and ω' is a non-negative weighting variable that represents the relative importance of the network vulnerability with respect to the overall network cost. Here, it is important to note that $\omega \neq \omega'$ in general, because $\nu' \leq 1$, while $\mu_{quad} \leq r^2$. As it is much easier to compute μ_{quad} than ν', this heuristic is easier to implement than the previous one.

4. Simulation results and discussion

Our simulations are done using the network topologies made available by the Rocketfuel project [15]. For our simulations, we make the following simplifying assumptions. First, we assume that all network links have infinite capacities ($d_l \rightarrow \infty\ \forall\, l \in E$). Next, we assume that all the network links have the same probability of being tapped ($p_l = p\ \forall\, l \in E$).

For each scenario, we first decide on the network parameters, including the number of input random processes to be transmitted (r), the throughput rate for each process (ρ), the number of processes of interest to the wiretapper (k) and the number of sink nodes ($|T|$). We also decide on the network to be used, which gives us the c_l values.

We then proceed to run simulations with the above network parameters, and calculate the mean network cost and network vulnerability. For each simulation, each source node in S is randomly and uniformly chosen from V with replacement, before each sink node in T is randomly and uniformly chosen from $V - S$ without replacement. During this process, we also ensure that the multicast problem is feasible, by requiring the presence of at least one path from each source node to each sink node.

Following this, we let p vary from 0 to 0.1, and for each value of p, we proceed to solve for five different multicast subgraphs. We first consider the single path routing and the multipath routing solutions, before going on to consider the single path network coding solution by having the objective function in (1) as μ. Finally, we consider the two heuristics proposed in this paper.

For each subgraph, we estimate the network vulnerability in the following way. We randomly generate 2000 sets of tapped links based on p, and decide for each set, if it enables

Table 2: Network vulnerabilities for three different networks, with $r = 4, \rho = 10, k = 1, |T| = 4$, and $p = 0.01$ or 0.1.

	Exodus		Telstra		Tiscali	
	$p = 0.01$	$p = 0.1$	$p = 0.01$	$p = 0.1$	$p = 0.01$	$p = 0.1$
Single path routing	0.11	0.67	0.084	0.59	0.090	0.62
Multipath routing	0.025	0.39	0.044	0.45	0.047	0.47
Single path coding	0.054	0.62	0.060	0.59	0.083	0.69
Heuristic 1 ($\omega = 3 \times 10^5$)	0.012	0.57	0.028	0.55	0.023	0.60
Heuristic 2 ($\omega' = 300$)	0.013	0.73	0.024	0.60	0.068	0.86

the wiretapper to retrieve all processes in $\mathscr{W}_{interest}$. The sample mean is then taken to be the network vulnerability (ν).

From Table 2, we observe that when $p = 0.01$, not only are the values of ν associated with the two heuristics comparable to that of multipath routing, but they are often the lowest values out of the five. However, when p increases to 0.1, both heuristics fail to match up to multipath routing in terms of the network vulnerability. In fact, we observe that the second heuristic yields the highest network vulnerability.

To explain these observations, we consider the coded network shown in Figure 2(b) where the wiretapper is interested in X_1. When p is very small, we can approximate the situation as one where the wiretapper has negligible probability of tapping two or more network links. Thus, the only way for him to retrieve X_1 is by tapping one of the two links that are carrying X_1 in an uncoded fashion. For a similar routing strategy on the network, at least four network links must be used for the multicast of X_1, and the tapping of any one of these links will enable the retrieval of X_1. Thus, routing results in a higher value of ν than network coding. In general, when p is small, better security is achieved by network coding as $\mathscr{Y}_{tapped}$ is often small, and the wiretapper is unable to retrieve enough degrees of freedom to decode $\mathscr{W}_{interest}$ entirely.

In routed networks, we first note that the tapping of links carrying messages outside $\mathscr{W}_{interest}$ does not help the wiretapper in the retrieval of the messages in $\mathscr{W}_{interest}$. However, in coded networks, the tapping of such links can potentially aid the wiretapper. Consider the network in Figure 2(b), although $H(X_1 \mid X_1 + X_2) > 0$ and $H(X_1 \mid X_2) > 0$, we note that $H(X_1 \mid \{X_1 + X_2, X_2\}) = 0$. In general, for coded networks, there are more ways for the wiretapper to decode the messages in $\mathscr{W}_{interest}$ from $\mathscr{Y}$. This reduces the security advantage of transmitting encoded messages, and results in the higher values of ν seen in coded networks when p is large.

Figure 3 shows the plots of two possible functions that $\mathscr{F}(\mu, \nu)$ can take, for the Exodus network. Figures 3(a) and 3(b) show the plots for $\mathscr{F}(\mu, \nu) = \mu + \omega\nu$, where $\omega = 3 \times 10^5$ (the value of ω used in the first heuristic). Figures 3(c) and 3(d) show the plots for $\mathscr{F}(\mu, \nu) = \mu\nu$.

From Figures 3(a) and 3(b), we see that the first heuristic performs the best out of the three coding strategies for all values of p, but the second heuristic performs quite well when p is small. In addition, the routing cases perform worse than the coding cases for small values of p, mainly because of the higher network cost incurred in routed networks. From Figures 3(c) and 3(d), we again observe that the coding schemes perform better than the routing schemes for small values of p.

Simulations done on the Telstra and Tiscali networks show similar results. In particular, we see that for the case where $\mathscr{F}(\mu, \nu) = \mu\nu$, both routing solutions returned higher values of $\mu\nu$ than the coding solutions for all values of p between 0 and 0.1. The corresponding plots for the Telstra and Tiscali networks are shown in Figures 4(a) and (b), respectively.

Figure 5 shows the plot of network vulnerability against p for the first heuristic, but with different values of r. The values of ω were appropriately scaled to compensate for the higher values of μ with increasing r.

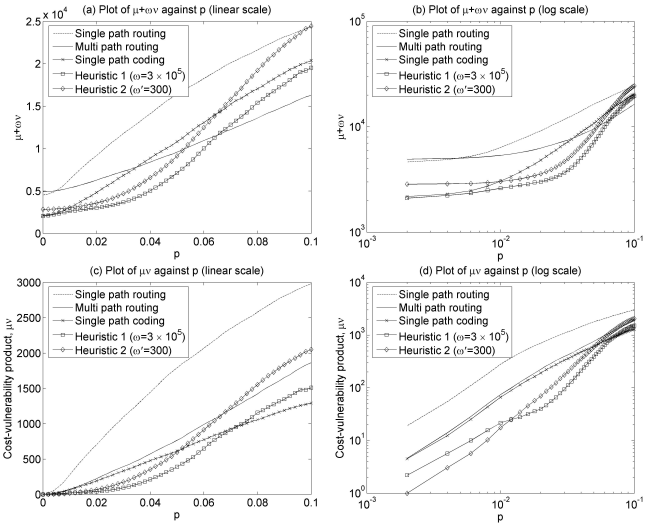

Figure 3: Plot of two possible realizations of $\mathscr{F}(\mu, \nu)$ against p. (Exodus network, $r = 4, \rho = 10, k = 1, |T| = 4$).

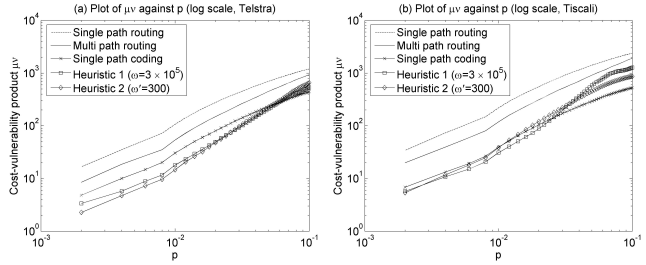

Figure 4: Plot of $\mu\nu$ against p for the Telstra and the Tiscali networks. ($r = 4, \rho = 10, k = 1, |T| = 4$).

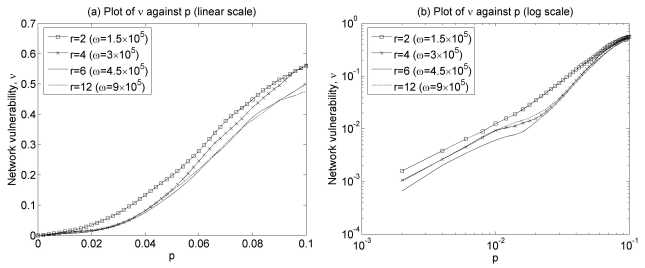

Figure 5: Plot of ν against p for different values of r (first heuristic). (Exodus network, $\rho = 10, k = 1, |T| = 4$).

From Figure 5, it is observed that when r increases from 2 to 6, ν decreases, but when r increases further to 12, ν does not decrease anymore. On the contrary, ν increases for small values of p when $r = 12$. This interesting behavior can be explained as follows.

When r increases, it affects the network vulnerability in several ways. First, the mean number of encoded processes X_i in the transmitted processes $Y(l')$ grows as $O(r)$. This decreases ν as the wiretapper will, on average, need to retrieve more degrees of freedom from the network in order to decode all the messages in $\mathscr{W}_{interest}$. Second, the size of $\mathscr{Y}$ grows as $O(2^r)$, increasing the number of ways $\mathscr{W}_{interest}$ can be decoded from the messages in $\mathscr{Y}$, thus increasing ν. Finally, the average amount of flow on the network links grows as $O(r)$. Thus, for any fixed p, the expected size of $\mathscr{Y}_{tapped}$ increases, and ν increases.

As the increase in r can affect the network vulnerability in these different ways, the net effect of an increase in r on ν is ambiguous. However, the simulation results suggest that when r is small, an increase in r decreases ν, while for larger values of r, an increase in r increases ν. This is probably due to the different growths of the network parameters mentioned in the previous paragraph. As the size of $\mathscr{Y}$ grows as $O(2^r)$, when r is small, the growth of $\mathscr{Y}$ is small, and the net effect of an increase in r is a decrease in ν. However, when r gets bigger, the growth of $\mathscr{Y}$ becomes much larger, resulting in a net increase in ν.

5. Conclusion

In this paper, we have considered the problem of providing a set of connections at both a low cost and a high level of security against wiretapping. As an exact solution is difficult, we have presented two heuristic methods of finding the coding subgraphs that can achieve this. Through our simulation results, we observe that a trade-off between network cost and network vulnerability also exists in coded networks. In addition, we observe that network coding can be more effective than traditional routing for low cost and secure data multicast, especially when the links are not too easily tapped.

In this study, we have focused on the problem of finding the coding subgraph, and conducted our simulations in a centralized manner. However, it should be noted that the finding of the coding subgraph can done in a decentralized manner for the second heuristic that we have proposed, since it has a strictly convex cost function for each network link [3]. As we have assumed the use of a random linear network code, which itself is implementable in a distributed manner, we conclude that our second heuristic solution to the problem can be implemented in an entirely decentralized manner.

As we have only considered the issue of resilience against wiretapping in this work, future research can be done to investigate the other security issues of network coding. Furthermore, as we have only suggested two simple heuristic approaches to the problem, alternate algorithmic approaches remain as clear avenues for future work.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204 - 1216, Jul. 2000.

[2] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," *IEEE International Symposium on Information Theory and its Applications*, 2004.

[3] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," *Proc. - IEEE INFOCOM*, Mar. 2005.

[4] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[5] N. Cai and R. W. Yeung, "Secure network coding," *Proc. - IEEE International Symposium on Information Theory*, pp. 323, 2002.

[6] J. Feldman, T. Malkin, R. A. Servedio and C. Stein, "On the capacity of secure network coding," *Proc. - 42nd Annual Allerton Conference on Communication, Control and Computing*, September 2004.

[7] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," *Proc. - IEEE International Symposium on Information Theory*, pp. 144, 2004.

[8] J. Jeong, and J. Ma, "Security analysis of multi-path routing scheme in ad hoc networks," *Lecture Notes in Computer Science*, vol. 3320, pp. 694 - 697, 2004.

[9] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," *Proc. - IEEE INFOCOM*, vol. 4, pp. 2404 - 2413, 2004.

[10] J. Yang, and S. Papavassiliou, "Improving network security by multipath traffic dispersion," *Proc. - IEEE MILCOM*, vol. 1, pp. 34 - 38, 2001.

[11] C. K.-L. Lee, X.-H. Lin, and Y.-K. Kwok, "A multipath ad-hoc routing approach to combat wireless link insecurity," *IEEE International Conference on Communications*, vol. 1, pp. 448 - 452, 2003.

[12] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371 - 381, Feb. 2003.

[13] T. Ho, M. Médard, J. Shi, M. Effros and D. R. Karger, "On randomized network coding," *Proc. - 41st Annual Allerton Conference on Communication Control and Computing*, Oct. 2003.

[14] R. Koetter, and M. Médard, "Beyond routing: An algebraic approach to network coding," *Proc. - IEEE INFOCOM*, vol. 1, pp. 122 - 130, 2002.

[15] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Trans. Networking*, vol. 12, no. 1, pp. 2 - 16, Feb. 2004.