Power-Aware Traffic Cover Mode to Prevent Traffic Analysis in Wireless Ad Hoc Networks

Shu Jiang, Nitin H. Vaidya, and Wei Zhao

Abstract— **Traffic analysis poses a serious threat to communication security, especially in wireless networks. Encryption may be used to hide message *contents*, whereas traffic *padding* may be used to hide the traffic *pattern*. This paper deals with the issue of preventing traffic analysis by inserting dummy (or padding) traffic to hide the real traffic pattern. The observable traffic pattern, after dummy traffic has been inserted, is referred to as a cover mode. Since the dummy traffic incurs an overhead, it is important to minimize dummy traffic while achieving the desired security objective. In this paper, we attempt to develop a suitable cover mode with the objective of minimizing energy consumption – the nodes in wireless ad hoc networks typically have limited battery supply, therefore, it is important to minimize the energy consumption. We propose different methods for implementing the cover mode, and evaluate the impact of these techniques on system lifetime and on energy consumption per unit of real traffic transmitted in the network.**

Keywords—**Security, Information Hiding, Traffic Analysis, Cover Mode, Wireless Ad hoc Network, Power Consumption.**

I. Introduction

Wireless ad hoc networks can be formed by a group of computers using peer-to-peer wireless links, without making use of any pre-existing infrastructure [1]. Due to their ease of deployment, ad hoc networks are suitable for many civilian as well as military applications. It is well known that the wireless medium provides greater opportunities for eavesdropping on the transmissions, as compared to a wired network. Many approaches can be used to combat this problem – one class of techniques makes it difficult to eavesdrop on wireless transmissions (e.g., frequency-hopping using a secret hopping pattern), and another class of techniques makes it difficult to learn useful information even if an eavesdropper can hear the wireless transmissions. This paper is related to the latter approach. When an intruder can hear wireless transmissions, two types of information may potentially be obtained:

- Contents of the transmitted packets: To prevent an intruder from learning the contents, the transmitted packets are encrypted using a secret key. This paper assumes that such an encryption mechanism is being used.
- Traffic pattern: Even with encryption, an intruder can potentially learn useful information by observing the traffic pattern. Information may potentially also be learned by observing when a *change* in the traffic pattern occurs. This type of security threat is referred to as the *traffic analysis* threat.

  To prevent traffic analysis, it is important to hide not only the real traffic pattern, but also the changes in the real traffic pattern. This paper considers algorithms for preventing traffic analysis in wireless ad hoc networks, with the goal of keeping energy consumption low.

Previous studies show that traffic padding and traffic rerouting are two effective countermeasures against the traffic analysis threat [2], [3], [4]. When traffic padding is applied, dummy data packets are injected into the network. With appropriate encryption, the eavesdropper cannot distinguish dummy packets from real packets. So the traffic pattern observed by an eavesdropper is not the pattern of real traffic. Another technique for manipulating the traffic pattern is traffic rerouting – in the rerouting technique, traffic between an origin-destination pair is split and delivered along several routes [3], [4].

In this paper, we consider the traffic padding technique. We do not consider the approach of routing traffic from a single flow on multiple routes, due to the potential difficulties created by reordering of packets.

We consider the situation wherein the network may operate in any one of many possible *operation modes* (each operation mode has a distinct traffic pattern). A *cover mode* is constructed such that the intruder cannot determine the real operation mode of the network at any given time. In addition, to make it difficult to detect the change in operation mode, our schemes use an identical cover mode for all operation modes. We propose several schemes for constructing such a cover mode and evaluate their performance. Specifically, as elaborated later, we consider *end-to-end* as well as *link-level* approaches for padding traffic.

Insertion of padding traffic in the network leads to higher energy consumption. We consider two metrics to characterize energy consumption with and without traffic padding: the first metric measures the system lifetime, and the second metric is the average energy consumption per unit of useful data delivered by the network. These metrics are explained later in the paper.

The rest of the paper is organized as follows. Section II presents our network model and discusses the two schemes of constructing cover mode. In Section III, we give the formal definition of performance metrics and formulate the optimal cover mode problem. Heuristic solutions are proposed in Section IV. Section V presents our results of simulations evaluating the power-awareness of different cover modes. Finally, Section VI summarizes the main results.

II. Preliminaries

The ad hoc network under consideration can be represented by a graph. In general, the route between a pair of nodes may consist of multiple hops through several intermediate nodes.

An *operation mode* of the network is defined by a set of *end-to-end flows* and the *traffic* imposed by each of these flows. An

The authors are with Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA. E-mail: {jiangs, vaidya, zhao}@cs.tamu.edu.

This research is sponsored by DARPA under contract number F30602-99-1-0531. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U.S. government.

end-to-end flow is characterized by (source, destination) tuple and its traffic demand – for brevity, we may refer to an end-to-end flow simply as a *flow*.

Two operation modes may either differ in the sets of flows that are active in those modes, or in traffic demands posed by the flows (even if the sets of flows active in the two modes are identical).

Many methods may be used to characterize the traffic imposed by a flow. For simplicity, we characterize traffic imposed by each end-to-end flow using its average rate.

Route taken by all packets in a flow is assumed to be fully defined as a function of source and destination identities – thus, all packets from a given source to a given destination traverse the same route. This type of routing is provided, for instance, by the Dynamic Source Routing (DSR) protocol [5] for ad hoc networks.

Although, in general, nodes in an ad hoc network would be mobile, in this paper, we consider static network topologies. The problem of preventing traffic analysis in a network whose topology dynamically changes is strictly harder than that on a fixed topology – the work reported here constitutes the first step towards obtaining a general-purpose solution for mobile ad hoc networks.

Although transmissions from a given node to all its neighbors share the same wireless channel, for the ease of discussion, we will say that packets from node X to its neighbor Y traverse "link" (X,Y). A *link flow* on link (X,Y) is the aggregate of all end-to-end flows whose routes include link (X,Y). Thus, *link traffic rate* of a link is the sum of the rates of all end-to-end flows traversing that link.

We assume that all transmitted packets are encrypted using a key that is unknown to eavesdroppers. Therefore, even if transmitted packets are intercepted by an eavesdropper, the message contents encapsulated in the packets are not disclosed.

It is also ensured that, although the authorized nodes can distinguish a dummy packet from a real packet, an eavesdropper cannot make this determination – to achieve this, a dummy packet would have to be tagged with a special flag (which itself should also be encrypted).

We now define two types of cover mode, and qualitatively compare their advantages and disadvantages.

A. End-to-End Cover Mode

An end-to-end cover mode maintains a constant rate of traffic between each (source, destination) pair, independent of the actual operation mode. Consider two operation modes, with traffic rate B_1 and B_2, respectively, for the flow from node s to node d. An end-to-end cover mode must keep the traffic rate from s to d constant regardless of the operation mode. Thus, in the end-to-end cover mode, traffic rate from node s to d must be at least $\max(B_1, B_2)$. Without loss of generality, let us assume that $B_1 < B_2$. Also, let us assume that the cover mode maintains traffic of $\max(B_1, B_2)$, which is equal to B_2 (if $B_1 < B_2$). Thus, when the network is operating in the first operation mode, padding traffic must be inserted by node s at the rate of $B_2 - B_1$.

For future reference, observe that the dummy packets (i.e., padding traffic) are inserted by the source node for each flow. These packets traverse the entire route taken by the flow. The flow destination identifies the dummy packets and discards them on receipt. The rate at which dummy packets are inserted is independent of the *route* taken by the flow. Also, intermediate nodes on the flow's route need not take any special actions to implement the cover mode.

Recall that packets belonging to a given flow always traverse the same route. Thus, when an end-to-end cover mode is used, in addition to the end-to-end flows maintaining constant traffic rates, the *link traffic rate* for each link in the ad hoc network also remains constant, independent of the operation mode.

B. Link Cover Mode

A link cover mode is obtained by achieving constant traffic rate on each link in the ad hoc network, independent of the actual operation mode. As observed above, an end-to-end cover mode also yields constant link traffic rates – thus, an end-to-end cover mode is also a link cover mode (however, the converse is not always true).

Unlike the end-to-end cover mode, link cover mode is implemented by inserting dummy packets on each link, so as to maintain a constant rate on that link (recall that, to implement the end-to-end cover mode, the flow source inserts dummy traffic). Thus, for link (X,Y), node X may need to insert dummy packets to achieve the link cover mode. Observe that node X may potentially be neither the source nor the destination for any flow carried on link (X,Y).

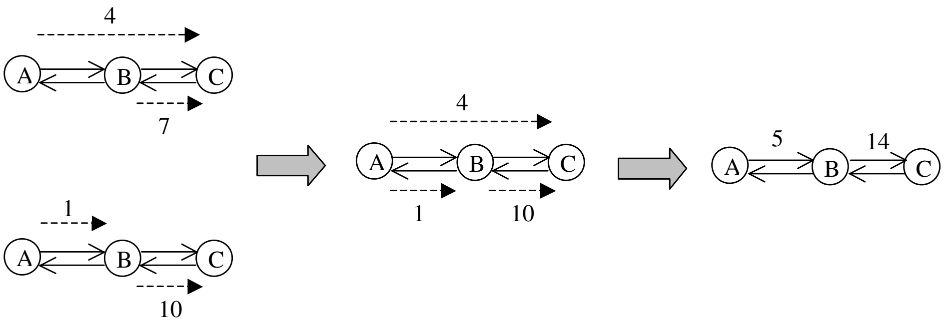
Note that the link cover mode is a function of routes used for the end-to-end flows. This is unlike the end-to-end cover mode (which can be determined independent of the chosen routes). On the other hand, although the link traffic rate on each link is constant in the link cover mode (independent of the real operation mode), the end-to-end flows may have different rates in different operation modes.

In the link cover mode, dummy traffic is inserted on a per-link basis, the node at the receiving end of the link – node Y for link (X,Y) – must identify and remove the dummy traffic received on link (X,Y). (Recall that when using the end-to-end cover mode, only the destination node of a flow has to be able to identify the dummy packets). Thus, for the link cover mode, a per-link encryption mechanism must be used so that: (a) the link receiver can distinguish dummy packets from the real packets, but an eavesdropper cannot, and (b) an eavesdropper cannot determine the end-to-end (source, destination) for a given packet.

In general, a link cover mode can be implemented by inserting a smaller amount of dummy traffic, as compared to implementing an end-to-end cover mode. However, the disadvantage of the link cover mode is that it is a function of chosen routes, and dummy packets must be inserted (and removed) on each link. Thus, both approaches have their advantages and disadvantages.

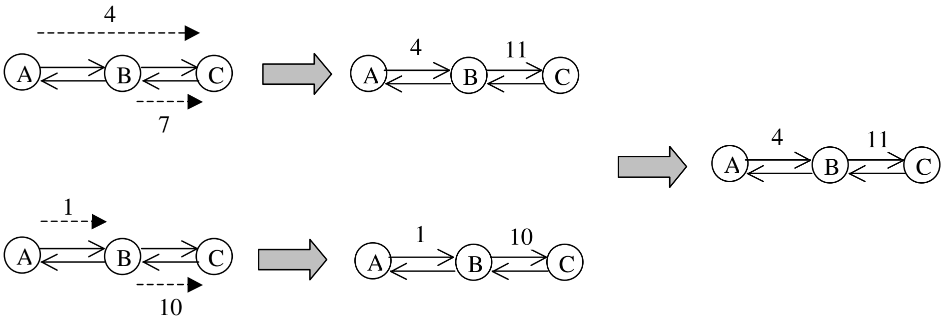
We now illustrate the difference between end-to-end and link cover modes using an example.

Example: Consider the network of 3 nodes, illustrated in Fig. 1 and 2. Assume that this network has two operation modes. Mode 1 consists of one flow from node A to C at the rate of 4 data units/sec, and another flow from B to C at the rate of 7 data units/sec. Mode 2 consists of a flow from node A to B at the rate of 1 data unit/sec and a flow from node B to C at the rate of 10



(a) operation modes (b) end-to-end cover mode (c) link traffic rate

Fig. 1. Produce end-to-end cover mode



(a) operation modes (b) link traffic rate in operation modes (c) link cover mode

Fig. 2. Produce link cover mode

data units/sec. Fig. 1(a) shows the two operation modes.

From the prior discussion, it follows that an *end-to-end cover mode* containing three flows can be obtained as follows: (i) a flow from A to B at 1 data unit/sec, (ii)a flow from A and C at 4 data units/sec, and (iii) a flow from B and C at 10 data units/sec. The end-to-end cover mode is illustrated in Fig. 1(b). Fig. 1(c) illustrates the link traffic rates obtained for this end-to-end cover mode, taking into account the routes taken by the flows. Observe that there is traffic with rate 5 data units/sec on link (A, B) and with rate 14 data units/sec on link (B, C).

Now, Fig. 2 illustrates a link cover mode for the same two operation modes, which are also shown in Fig. 2(a). To obtain the link cover mode, we first determine the link traffic rate for each of the operation modes, taking into account the routes used by these flows. The link traffic rates for the two modes are shown in Fig. 2(b). Fig. 2(c) illustrates the *link cover mode* for this example – link traffic rate for a link, say (X,Y), in this link cover mode is obtained as the maximum link traffic rate on the link over all operation modes. For instance, traffic rate on link (A, B) is 4 data units/sec in mode 1 and 1 data unit/sec in mode 2. Thus, in the link cover mode, the link traffic rate is $\max(4, 1) = 4$ (data units/sec).

Comparing Fig. 1(c) and Fig. 2(c), observe that, the link cover mode has a lower bandwidth requirements than the end-to-end cover mode.

In general, when using either of the above two approaches, a cover mode introduces additional traffic in the network (as compared to the operation mode) which consumes additional energy. Since energy resource is often restricted in wireless hosts, utility of a security scheme depends on the extent to which it affects the energy requirements. In the next section, we define metrics which we use to characterize the energy consumption, and define the optimal cover mode problem.

III. Problem definition

We model the network as a directed graph $G(N, L)$ where N is the set of all nodes and L is a set of directed links. Link (i, j) is in L (where $i, j \in N$) if node j can receive packets from node i. Define a set S_i for each node i, such that node $j \in S_i$ if and only if $(i, j) \in L$. We also define the following notations:

W	a set of end-to-end flows
r_w	average traffic rate of an end-to-end flow w
P_w	set of all candidate routes for an end-to-end flow w.
$x_{p,w}$	routing decision variable which is 1 if path p is used for end-to-end flow w and 0 otherwise.
$\delta_{ij}(p)$	indicator function which is 1 if link (i, j) is on path p and 0 otherwise.
λ_{ij}	traffic rate on link $(i, j) \in L$
E_i	initial energy level at node $i \in N$
τ	energy consumed when a node transmits unit data
ρ	energy consumed when a node receives unit data
e_i	rate of energy consumption at node $i \in N$
T_{sys}	system lifetime: System lifetime is the duration of time which elapses before at least one node is depleted of energy.

$\bar{e}$ average energy consumption per unit of data delivered to its destination. Calculation of $\bar{e}$ takes into account energy consumed by all nodes forwarding a packet.

When we need to identify parameters that relate to a particular mode, superscript (i) is added to the notation representing mode i and superscript C is added representing cover mode. For example, $W^{(1)}$ denotes the set of end-to-end flows in mode 1 and λ_{ij}^{C} denotes the traffic rate on link (i,j) in a cover mode. Whether end-to-end cover mode or link cover mode is referred will be clear by the context.

Recall that we assume that all packets belonging to an end-to-end flow are routed along the same path. Therefore, $\sum_{p\in P_w} x_{p,w} = 1$ for any $w \in W$. Also, we obtain the following equations:

$$\lambda_{ij} = \sum_{p,w:p\in P_w, w\in W} x_{p,w} r_w \delta_{ij}(p) \qquad (1)$$

and

$$e_i = \tau \sum_{j\in S_i} \lambda_{ij} + \rho \sum_{j:i\in S_j} \lambda_{ji} \qquad (2)$$

Equation (1) is based on the observation that traffic rate on any given link is the sum of the traffic rates of all end-to-end flows carried by that link. Equation (2) states the fact that the total energy consumption by each node has two parts: the energy consumed when transmitting packets, and energy consumed when receiving packets. Compared to receiving, the energy required for transmitting same amount of data is typically much higher. For instance, the DEC Roamabout radio consumes approximately 5.76 watts during transmission and 2.88 watts during receiving [6]. A 915MHz Lucent WaveLAN PCMCIA wireless Ethernet card requires 3.00 watts when transmitting and 1.48 watts when receiving [7].

We base our definition of *system lifetime* on that presented in [8]. System lifetime T_{sys} is defined as the duration of time until at least one node drains out its energy reserve. Thus,

$$\begin{aligned} T_{sys} &= \min_{i\in N} \frac{E_i}{e_i} & (3)\\ &= \min_{i\in N} \frac{E_i}{\tau\sum_{j\in S_i}\lambda_{ij} + \rho\sum_{j:i\in S_j}\lambda_{ji}} & (4)\end{aligned}$$

Note that, while our definition of T_{sys} attempts to take into account both energy consumption when transmitting a packet as well as when receiving a packet, [8] considers energy consumption only when transmitting packets. So our definition is somewhat more accurate than [8] – nevertheless, our definition (similar to [8]) ignores the power consumed by nodes when they overhear transmissions not intended for them. Our energy consumption model is more accurately applicable for power-conserving MAC protocols such as PAMAS [6], which turn off radios when not actively needed to receive or transmit a packet.

Observe that T_{sys} is a function of the operation mode when a cover mode is not used. When a cover mode is used, T_{sys} depends on the flow traffic rates in the cover mode – note that when a cover mode is used, T_{sys} does not depend on the actual operation mode used. In general, T_{sys} will reduce when a cover mode is used (as compared to T_{sys} in an operation mode), therefore, in the performance evaluation section we measure the percentage reduction in T_{sys} – since the original T_{sys} is a function of the operation mode, the percentage reduction when compared to each operation mode may be different (as seen later in the paper).

In addition to T_{sys}, we define another metric $\bar{e}$ to measure the average energy consumption by the system to deliver an unit of useful data to its destination. For an operation mode characterized by flows in set W, $\bar{e}$ is calculated as

$$\begin{aligned} \bar{e} &= \frac{\sum_{i\in N} e_i}{\sum_{w\in W} r_w} & (5)\\ &= \frac{\sum_{i\in N}(\tau\sum_{j\in S_i}\lambda_{ij} + \rho\sum_{j:i\in S_j}\lambda_{ji})}{\sum_{w\in W} r_w} & (6)\end{aligned}$$

Similar to T_{sys}, when a cover mode is not used, $\bar{e}$ is a function of the operation mode. However, unlike T_{sys}, when a cover mode is used, $\bar{e}$ is a function of the cover mode *as well as* the actual operation mode. For instance, when network is in a cover mode but its real operation mode is mode 1, , $\bar{e}$ is given by

$$\begin{aligned} \bar{e} &= \frac{\sum_{i\in N} e_i^C}{\sum_{w\in W^{(1)}} r_w} & (7)\\ &= \frac{\sum_{i\in N}(\tau\sum_{j\in S_i}\lambda_{ij}^C + \rho\sum_{j:i\in S_j}\lambda_{ji}^C)}{\sum_{w\in W^{(1)}} r_w} & (8)\end{aligned}$$

Analogous to the percentage decrease in T_{sys}, in our performance evaluation, we measure the percentage increase in $\bar{e}$ when a cover mode is used.

The two methods for defining cover mode, and the above two metrics, give rise to four optimality problems defined below.

A. End-to-end cover mode maximizing T_{sys}

Specification of an optimal cover mode includes two components: (i) the set of active flows and their traffic rates, and (ii) routes used for the active flows. The first problem is relatively simple – the rate of flow w in the optimal end-to-end cover mode may be obtained simply as a maximum rate of flow w over all modes – it is easy to see that any cover mode that uses flow traffic rates higher than this will necessarily result in higher energy consumption. (Note that if flow w is not active in a certain operation mode, its rate is essentially 0 in that mode.)

Given the flow rates in the optimal end-to-end cover mode, the problem of finding suitable routes for the flows can be stated as follows:

$$Maximize \quad T_{sys} \qquad (9)$$

subject to:

$$T_{sys} * (\tau \sum_{j\in S_i} \lambda_{ij}^C + \rho \sum_{j:i\in S_j} \lambda_{ji}^C) \le E_i, \quad \forall i \in N \qquad (10)$$

$$\lambda_{ij}^{C} = \sum_{p,w:p\in P_w, w\in W^C} x_{p,w} r_w \delta_{ij}(p), \quad \forall (i,j) \in L \tag{11}$$

$$\sum_{p\in P_w} x_{p,w} = 1, \quad \forall w \in W^C \tag{12}$$

and

$$x_{p,w} = 0 \ \ or \ \ 1, \quad \forall p \in P_w \ \forall w \in W^C \tag{13}$$

where W^C is the set of end-to-end flows in the end-to-end cover mode.

The problem above is a 0-1 integer linear programming problem, which is NP-complete [9]. There are no polynomial algorithms for solving this problem optimally. In Section IV, we present a heuristic algorithm based on the flow deviation (FD) method [10], [11] to find a locally optimal solution.

B. End-to-end cover mode minimizing $\bar{e}$

As for the previous problem, the rate of flow w in the optimal end-to-end cover mode is obtained simply as the maximum rate of flow w over all operation modes.

Now, we only need to address the issue of finding suitable routes so as to minimize $\bar{e}$, given the flow rates in the cover mode. From the definition of $\bar{e}$, it follows that minimizing $\bar{e}$ is equivalent to minimizing the total energy consumption by all nodes in the network. Observe that, if an end-to-end flow w has rate r_w, then, *at each hop* on its path, the transmission of the packets contributes energy consumption at the rate $(\tau + \rho) * r_w$ (this includes energy consumed at transmitting and receiving endpoints of the link). Therefore, the path with the minimal number of hops, i.e. the shortest hop path, is the optimal path for w. Actually, for any mode, routing all end-to-end flows along their respective shortest hop paths produces the minimal value of $\bar{e}$. Thus, the problem of finding an end-to-end cover mode while minimizing $\bar{e}$ reduces to the shortest path routing problem (where length of a path is the number of hops on the path).

C. Link cover mode maximizing T_{sys}

In an optimal link cover mode, the traffic rate on each link is the maximal possible rate of real traffic on that link in any operation mode. Formally, if there are M operation modes in the network, the traffic rate on link (i,j) in the link cover mode is given by

$$\lambda_{ij}^{C} = \max_{m}(\lambda_{ij}^{(m)}), \quad m = 1, 2, \ldots, M$$

The problem of maximizing T_{sys} can be stated as follows:

$$Maximize \quad T_{sys} \tag{14}$$

subject to:

$$T_{sys} * (\tau \sum_{j\in S_i} \lambda_{ij}^{C} + \rho \sum_{j:i\in S_j} \lambda_{ji}^{C}) \leq E_i, \quad \forall i \in N \tag{15}$$

$$\lambda_{ij}^{C} = \max_{m}(\lambda_{ij}^{(m)}), \quad m = 1, 2, \ldots, M \tag{16}$$

$$\lambda_{ij}^{(m)} = \sum_{p,w:p\in P_w, w\in W^{(m)}} x_{p,w} r_w \delta_{ij}(p), \quad \forall (i,j) \in L \ \forall m \tag{17}$$

$$\sum_{p\in P_w} x_{p,w} = 1, \quad \forall w \in W^{(m)} \ \forall m \tag{18}$$

and

$$x_{p,w} = 0 \ \ or \ \ 1, \quad \forall p \in P_w \ \forall w \in W^{(m)} \ \forall m \tag{19}$$

where $W^{(m)}$ is the set of end-to-end flows in mode m.

The above problem is a 0-1 integer linear programming problem with nonlinear constraints. Thus it is difficult to solve the problem optimally. Again, we take the flow deviation method. A heuristic algorithm is proposed in Section IV(C).

D. Link cover mode minimizing $\bar{e}$

As end-to-end cover mode, minimizing $\bar{e}$ is equivalent to minimizing the total rate of energy consumption by all nodes in the network. For the same reason as in Section III(C), the problem of finding the optimal link cover mode with minimum $\bar{e}$ is an integer programming problem. However, unlike the case of end-to-end cover mode minimizing $\bar{e}$, for the link cover mode, the routes do not necessarily use paths with the smallest number of hops. We present a heuristic algorithm in Section IV(D).

IV. Routing algorithm

In this section, we present a heuristic routing algorithm based on flow deviation method[10], [11]. First, we briefly describe the flow deviation method. Then we show how this method can be used to cope with the problems defined in Section III.

A. Flow deviation method [10], [11]

Flow deviation (FD) method is a general method for solving nonlinear programming problems. A FD-based algorithm is divided into two phases: the *initialization* phase and the *updating* phase. In the initialization phase, an initial solution of the problem is found. In our case, to form the initial solution, we choose a shortest hop path for each end-to-end flow. Then the FD algorithm enters *updating* phase in which the initial solution is improved incrementally by changing the routing paths of some end-to-end flows. The rerouting of end-to-end flows is not made simultaneously; instead, one end-to-end flow is selected at one time whose path is established with the goal to optimize the objective function. Through coordinated rerouting process, a locally optimal routing solution can be reached ultimately.

FD-based Routing Algorithm

(1) Compute the shortest hop path for each end-to-end flow.
(2) Select an end-to-end flow w randomly or according to a sequence defined in advance. If the sequence is exhausted, it is simply repeated.
(3) Remove the traffic requirement r_w for w from the network.
(4) Fix the routing paths for all other end-to-end flows and reroute the selected end-to-end flow w aiming to optimize the objective function.
(5) If rerouting of flow w does not improve the objective function, restore its old routing path.
(6) Go to step (2) until all the end-to-end flows have been examined once, but no further improvements are possible.

In the worst case, the algorithm takes exponential time to converge. But it can be efficient in a probabilistic sense. [12] proves that the algorithm takes $O(n^2)$ iterations on average where $n = |N|$ is the number of nodes in the network.

B. Solution for the problem in Section III(A)

The FD-based algorithm is used to find an end-to-end cover mode with attempt to maximize the system lifetime. In the algorithm, the rerouting of the selected end-to-end flow w, whose source is node s and destination is node d, intends to find the optimal path maximizing the system lifetime T_{sys}. For a given path p from node s to node d, the would-be system lifetime – that is the system lifetime if path p were to be chosen to send packets from s to d – is

$$T_{sys} = \min(\frac{E_s}{e_{s,w} + \tau r_w}, \frac{E_d}{e_{d,w} + \rho r_w}, \quad (20)$$

$$\min_{k \in N_p, k \neq s, k \neq d} \frac{E_k}{e_{k,w} + (\tau + \rho) r_w}) \quad (21)$$

where N_p is the set of all nodes on path p and $e_{k,w}$ is the energy consumption rate by node k excluding any energy consumption by flow w.

Finding a path maximizing T_{sys} is equivalent to finding a path minimizing the following value:

$$\max(\frac{e_{s,w} + \tau r_w}{E_s}, \frac{e_{d,w} + \rho r_w}{E_d}, \max_{k \in N_p, k \neq s, k \neq d} \frac{e_{k,w} + (\tau + \rho) r_w}{E_k}) \quad (22)$$

because E_s, E_d and E_k all are constants here. If we define the weight of a path as in (22), then the path for w is the one with minimal weight value.

We may use Bellman-Ford algorithm [13] to find the optimal path. In the algorithm, when we check a new path from source node s to node i which is constructed by concatenating a known path from s to node j with link (j, i), the weight of the new path $p_{sj} + (j, i)$ is calculated by the following formula:

$$\max(Weight(p_{sj}), \frac{e_{j,w} + (\tau + \rho) r_w}{E_j}, \frac{e_{i,w} + \rho r_w}{E_i}) \quad (23)$$

We give the pseudo code for the algorithm as below, which finds the optimal path from source s to destination d for w.

Bellman-Ford Algorithm

(1) for each node $i \in N$
  $Weight(p_{si}) = 0$.

(2) for $k = 1$ to $|N| - 1$
  for each link $(j, i) \in L$
    if $Weight(p_{sj} + (j, i)) < Weight(p_{si})$,
    then
      replace p_{si} with path $p_{sj} + (j, i)$.

(3) return p_{sd}.

Step (1) initializes the variables. Step (2) performs the relax operation. After k-th iteration, the optimal path from source node s to each node with no more than k hops is found. Since there are no negative weight links, the algorithm will converge to the optimal path from node s to each node after $|N| - 1$ iterations.

C. Solution for the problem in Section III(C)

In this problem, the value of the objective function—system lifetime—is not only decided by the routing paths for the end-to-end flows in one mode but also that in all modes. So we extend the algorithm in Section IV(A) as follows. The incremental improvement on the initial solution is performed at two levels. At the mode level, we select one mode at one time and improve the routing paths for the end-to-end flows in the mode with routing paths for the end-to-end flows in other modes fixed. Then we switch to another mode and repeat the intra-mode improvement process. If all modes have been selected once and no improvements can be made, the algorithm stops. During the intra-mode optimization, the FD-based algorithm in Section IV(A) is used with one end-to-end flow examined at one time.

In the FD-based algorithm, the rerouting of a selected end-to-end flow w intends to find the optimal path that maximizes the system lifetime T_{sys}. Without loss of generality, w is assumed to be one of the end-to-end flows in mode 1, whose source is node s and destination is node d. Let $\bar{\lambda_{ij}}^{(1)}$ be the rate of real traffic on link $(i, j) \in L$ in mode 1 with the removal of r_w. Let $e_{k,w}$ be the energy consumption rate by node k in the cover mode after flow w being removed. If link (i, j) were to be on the path for w, then the traffic rate on (i, j) in the link cover mode would increase by

$$\Delta\lambda_{ij}^C = \max_m(\bar{\lambda_{ij}}^{(1)} + r_w, \lambda_{ij}^{(2)}, \ldots, \lambda_{ij}^{(m)}) - \max_m(\bar{\lambda_{ij}}^{(1)}, \lambda_{ij}^{(2)}, \ldots, \lambda_{ij}^{(m)}) \quad (24)$$

If a given path p from node s to node d were to be used, the system lifetime would be

$$T_{sys} = \min_{k \in N_p} \frac{E_k}{e_{k,w} + \tau \sum_{j \in S_k} \Delta\lambda_{kj}^C \delta_{kj}(p) + \rho \sum_{i: k \in S_i} \Delta\lambda_{ik}^C \delta_{ik}(p)}$$

where N_p is the set of all nodes on path p.

The problem of maximizing T_{sys} has a dual problem of minimizing the following value:

$$\max_{k \in N_p} \frac{e_{k,w} + \tau \sum_{j \in S_k} \Delta\lambda_{kj}^C \delta_{kj}(p) + \rho \sum_{i: k \in S_i} \Delta\lambda_{ik}^C \delta_{ik}(p)}{E_k} \quad (25)$$

So, if we assign each link $(i, j) \in L$ with weight $\Delta\lambda_{ij}^C$ and define the weight of a path as in (25), then the optimal path for w is the one with minimal weight value.

We propose the following extended Bellman-Ford (EBF) algorithm to find the optimal path for w. In original Bellman-Ford algorithm, at its k-th iteration, it identifies the optimal (in our context: minimal weight) path between the source node s and each node i, containing no more than k hops. EBF extends the Bellman-Ford algorithm by having each node i maintain a set of paths, $PATH(i)$. $PATH(i)$ records the optimal path from s to i through each input link of i. For instance, if $(v, i) \in L$, then the optimal path from s to i through (v, i) is recorded in $PATH(i)$ as $p_{si}(v)$, i.e., $PATH(i) = \{p_{si}(v) | i \in S_v\}$. During the relax operation, when we check a new path from source node s to node i which is constructed by concatenating a known path from s to node j in $PATH(j)$, $p_{sj}(u)$,with link (j, i), the weight of the new path $p_{sj}(u) + (j, i)$ is calculated by the following formula:

$$\max(Weight(p_{sj}(u)), \frac{e_{j,w} + \tau\Delta\lambda^C_{ji} + \rho\Delta\lambda^C_{uj}}{E_j}, \frac{e_{i,w} + \rho\Delta\lambda^C_{ji}}{E_i}) \tag{26}$$

This formula is based on the path weight definition and the observation that extending a path by one more link only adds to the power consumption by the sender and the receiver of the link. The pseudo-code for the algorithm is given below.

Extended Bellman-Ford (EBF) Algorithm

(1) for each node $i \in N$
  $PATH(i) = \emptyset$
  for each input link $(j, i) \in L$
    $Weight(p_{si}(j)) = 0$
(2) for $k = 1$ to $|N| - 1$
  for each link $(j, i) \in L$
    for each path $p_{sj}(u) \in PATH(j)$
      if $Weight(p_{sj}(u) + (j, i)) < Weight(p_{si}(j))$, then
        replace $p_{si}(j)$ with path $p_{si}(u) + (j, i)$.
(3) return $p_{sd}(i)$ whose weight is the maximal among all paths in $PATH(d)$.

The original Bellman-Ford algorithm takes $O(n^3)$ time where $n = |N|$ is the number of nodes in the network. In the Extended Bellman-Ford algorithm, the maximum number of input links of each node is n. So, the time complexity of EBF is $O(n^4)$.

Theorem 1: EBF algorithm returns the optimal path for w with minimal weight.

Proof: Proof is presented in Appendix A. ■

Solution for the problem in Section III(D)

The FD-based algorithm in Section IV(A) can be used to obtain a heuristic solution to this problem. The rerouting algorithm intends to find the optimal path for the selected end-to-end flow w with respect to the total energy consumption/sec by all nodes. Routing the packets belonging to w along a particular path may increase the traffic rates of the links on the path and hence increase the energy consumption rate by the nodes on the path. In Section III(C), we give the formula for calculating the would-be increase of traffic rate on a link in (24). An increase of traffic rate $\Delta\lambda^C_{ij}$ on link (i, j) contributes $(\tau + \rho) * \Delta\lambda^C_{ij}$ to the total energy consumption/sec. So, if we assign each link (i, j) with weight $\Delta\lambda^C_{ij}$, the most energy conserving path for w is the shortest path with path length defined as the sum of the weights of all links on the path.

In the next section, we evaluate the performance of cover modes obtained using the algorithms in this section.

V. Performance evaluation

We conducted extensive simulations to evaluate the performance of the proposed security schemes. The reduction in system lifetime (T_{sys}) and the increase in average energy consumption/data unit ($\bar{e}$) caused by implementing different cover modes were measured and compared.

For the simulation, we used the GT-ITM topology generator [14] software to generate connected random graphs. With the number of nodes given, a random graph was generated by adding a duplex link between any pair of nodes with a probability p. We varied the value of p from 0.1 to 0.5. Intuitively, $p = 0.1$ produces a sparse graph and $p = 0.5$ produces a dense graph.

At initial time, all nodes were assumed to have same energy conserve – thus E_i was independent of i. We assumed that $E_i = 10^7$ energy units. Also, we assumed that $\rho = 1$ energy unit/data unit and $\tau = 2$ energy units/data unit.

We generated two operation modes for the network as follows. For each flow in each mode, the endpoints were picked randomly. The flow rates were chosen uniformly distributed between 1 data unit/sec and 100 data units/sec. The number of flows in each mode is 300.

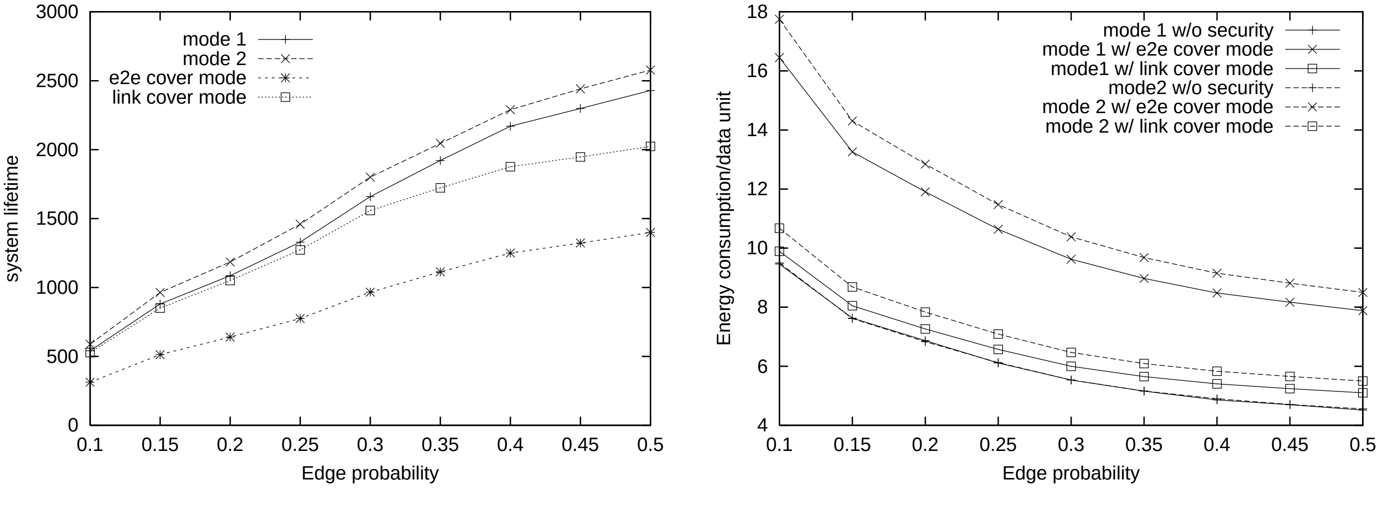
In each graph, we applied the algorithms in Section IV to acquire the value of T_{sys} and $\bar{e}$ in each mode with different security policies, without cover mode and with cover mode. In our implementation of the FD-based algorithm, flows were selected in sequence. Note that the attempt of maximizing T_{sys} and the attempt of minimizing $\bar{e}$ may end up with different routing solutions.

In the first experiment, we fixed the number of nodes in the network to 20 and changed the edge probability p to generate diverse random graphs. In Fig. 3, we plot the absolute values of T_{sys} and $\bar{e}$ in different cases. For each point in the figures, we ran the simulation for 20 times and plot the averaged result. As we mentioned earlier, the value of T_{sys} in a cover mode was independent of the real operation mode. But $\bar{e}$ exhibited different values when the real operation mode under cover changed, because $\bar{e}$ measured the unit energy consumption for delivering *useful* data.

We observe that when edge probability increases, the system lifetime is extended and the average energy consumption/data unit decreases in all cases. This is not surprising because in denser networks, more routes are available between each node pair for the system to choose from. So there is a better chance to get the initial routing solution improved.

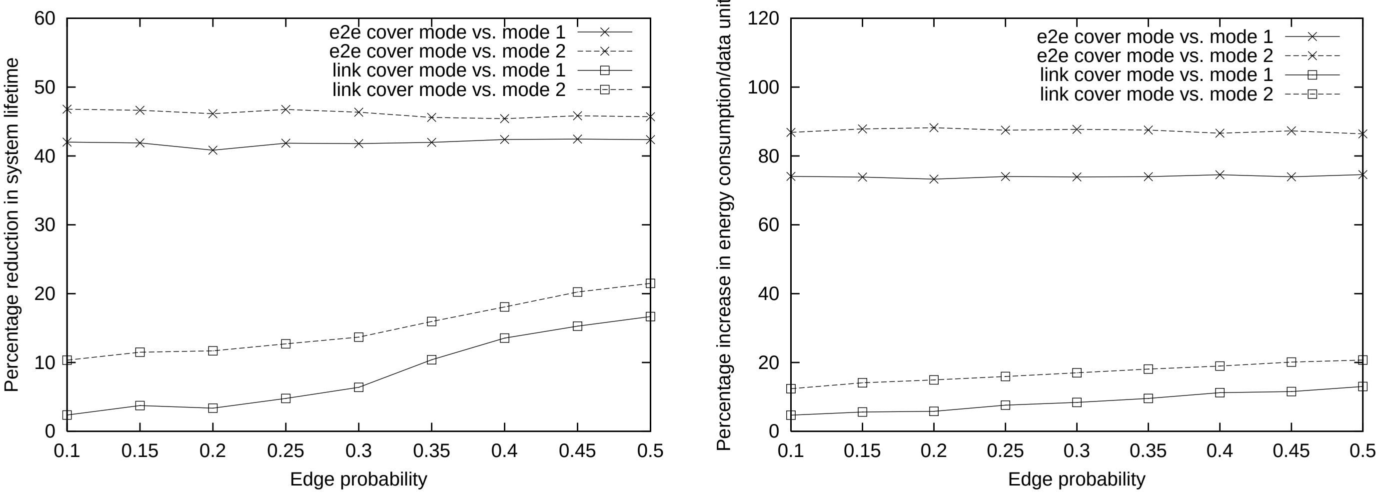
Our major concern is the performance degradation in the system after cover mode was implemented. Fig. 4 illustrates the percentage reduction in T_{sys} and the percentage increase in $\bar{e}$ after using different cover modes. We can see that if end-to-end cover mode was used, there would be a 40% reduction in the system lifetime or a 70% increase in the average energy consumption/data unit. On the other hand, the system lifetime reduction in link cover mode is in the 2-20% range and the increase in the average energy consumption/data unit is less than 20%. So, link cover mode is much more power conserving than end-to-end cover mode.

In our second experiment, we fixed two operation modes and changed the size of the network. We created random graphs all



(a) T_{sys} (b) $\bar{e}$

Fig. 3. Absolute values of performance metrics in 20-node random networks ($E_i = 10^7, \rho = 1, \tau = 2$)



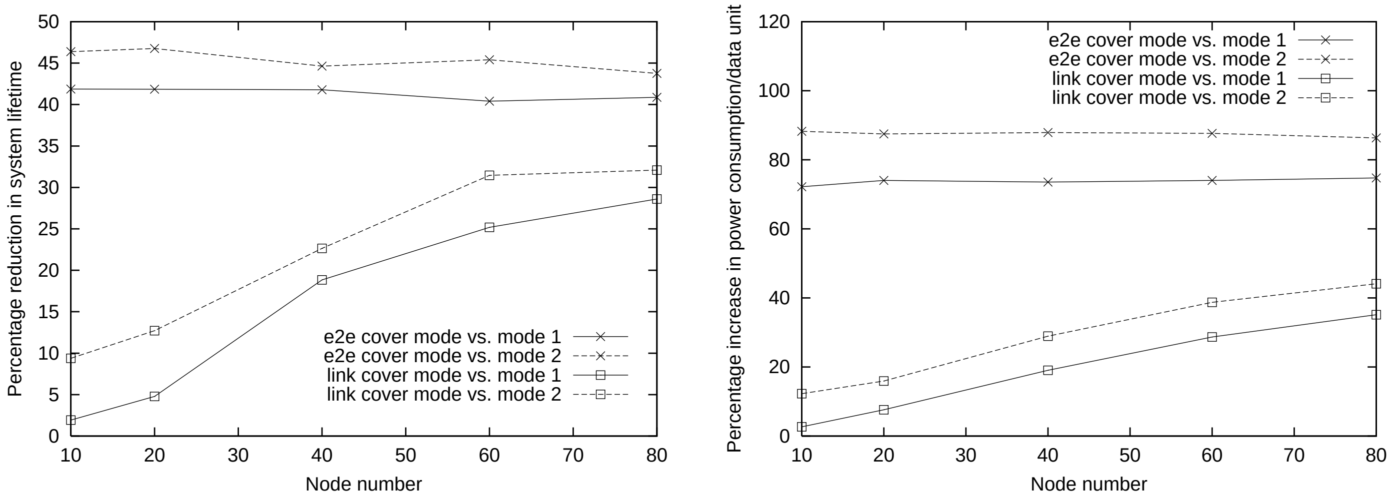
(a) T_{sys} (b) $\bar{e}$

Fig. 4. Percentage of performance degradation in 20-node random networks ($E_i = 10^7, \rho = 1, \tau = 2$)

with same edge probability ($p = 0.25$) but with different cardinalities ($|N| = 10, 20, 40, 60$ and 80). We were interested in how differently implementing cover mode affected the system performance in large networks from in small networks. The percentage reduction in T_{sys} and the percentage increase in $\bar{e}$ are demonstrated in Fig. 5.

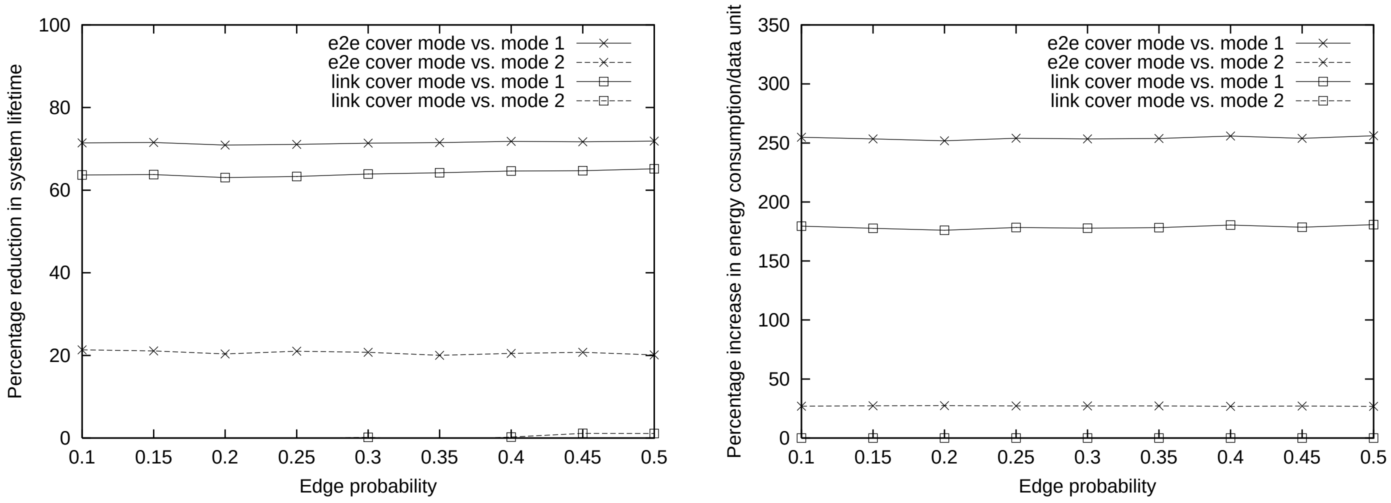
It is interesting to observe that when the network becomes "larger", the performance of end-to-end cover mode and link cover mode approaches each other. In other words, the link cover mode performs worse in larger networks than in smaller networks. Remember that when system searches for the "best" routing solution that leads to the "optimal" link cover mode, it shifts some end-to-end flows away from their shortest hop paths to reduce the traffic rates on some congested links. In larger networks, the alternative paths found for these flows are usually longer than the paths found in smaller networks with same edge probability, i.e. containing more nodes and links, which increases the possibility of perturbing the optimality of other existing paths. So the correlation among the flows will have more impact on the flow path selection in larger networks than it does in smaller networks.

In our third experiment, we generated two operation modes differently. The number of flows in each mode was still 300. But in mode 1, the flow rates were chosen randomly between 1 data unit/sec and 50 data units/sec while the maximum rate of a flow in mode 2 could be as high as 150 data units/sec. We measured the performance degradation caused by cover mode in 20-node networks with edge probability varied. Simulation results plotted in Fig. 6 show that the system lifetime achieved in link cover mode was even longer than that achieved in mode 2. Implementing link cover mode even produced a better performance than the original operation mode. This result was counter-intuitive. A likely reason is that our heuristic algorithm did not find the global optimal solution in some cases. A second observation we



(a) T_{sys} (b) $\bar{e}$

Fig. 5. Percentage of performance degradation in random networks with different number of nodes ($p = 0.25$, $E_i = 10^7$, $\rho = 1$, $\tau = 2$)



(a) T_{sys} (b) $\bar{e}$

Fig. 6. Percentage of performance degradation in 20-node random networks with operation modes generated differently ($p = 0.25$, $E_i = 10^7$, $\rho = 1$, $\tau = 2$)

can make is that the relative performance degradation of cover mode against mode 1 was much higher than that against mode 2, which implies that if the network spents most of the time in mode 1, the cost of preventing traffic analysis will be extremely high. But, if the network usually stays in mode 2 and switches to mode 1 occasionally, implementing a cover mode will be cost effective.

VI. Related work

Newman-Wolfe and Venkatraman conducted a study of preventing traffic analysis [3], [15], [16], [17]. Based on the end-to-end encryption assumption, they proposed a high level security scheme that combines traffic padding and host-based rerouting techniques. Guan et al. [4] inherited the model from Newman-Wolfe et al. and extended the high level security scheme to a real-time network where data delivery is subject to delay constraint. They developed algorithms for testing the feasibility of a cover mode in which the original traffic delay constraints are not compromised by enforcing security.

Radosavljevic and Hajek investigated the problem of hiding end-to-end flow traffic patterns in packet radio networks [18]. The solution they proposed is to develop a fixed node transmission schedule which is independent of any end-to-end flow traffic demands. The reduction of throughput by using this scheme was analyzed.

We presented some preliminary results related to the problem of traffic analysis in ad hoc networks in [19]. In the preliminary study, only the link cover mode was considered. Also, the performance metric used in [19] is different from the metrics used in this paper. This paper significantly extends on our preliminary work reported in [19].

Optimal routing problem has attracted persistent interests of researchers in decades. The techniques proposed have a wide spectrum. In addition to the flow deviation method we used in this paper, Gavish and Hantler [20] and Lin and Yee [21] all applied dual methods based on Lagrangean relaxation to solve this programming problem. Simulated annealing was first used in [12] to cope with this problem. In recent years, the optimal routing problem often took the form of quality-of-service(QoS) routing problem. A new class of routing problem has been identified with the optimal condition relaxed, i.e. *Multi-constrained path problem*(MCP) aiming to find feasible paths (not necessarily optimal) subject to multiple constraints (for example, on delay, cost, etc.). A number of heuristic algorithms have been proposed [22], [23], [24], [25], [26]. An extensive survey can be found in [23]. Proposal for incorporating QoS routing support into OSPF routing protocol has also been made by Guerin etc. [27].

VII. Conclusions

This paper studies the security problem of preventing traffic analysis in wireless ad hoc networks. Our approach inserts dummy traffic into the network so that an eavesdropper can learn the *cover mode* but not the actual operation mode. Two types of cover modes, end-to-end cover mode and link cover mode are examined in the paper. We proposed algorithms for finding cover modes which reduce energy consumption. Simulation results indicate that end-to-end cover mode generally performs worse than link cover mode, but in large networks, the two approaches yield similar energy overheads. It was also shown that when the traffic demands are skewed among the operation modes, implementing this security scheme may mean unacceptably high performance degradation.

References

[1] IETF Mobile Ad-Hoc Networks (MANET) Working Group, "http://www.ietf.org/html.charters/manet-charter.html," .

[2] O. Berg, T. Berg, S. Haavik, J. Hjelmstad, and R. Skaug, *Spread Spectrum in Mobile Communication*, IEEE, 1998.

[3] R. E. Newman-Wolfe and B. R. Venkatraman, "High level prevention of traffic analysis," in *Seventh Annual Computer Security and Applications Conference*, Dec. 1991.

[4] Y. Guan, C. Li, D. Xuan, R. Bettati, and W. Zhao, "Preventing traffic analysis for real-time communication networks," in *MilCom'99*, Oct. 1999.

[5] D. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and Hank Korth, Eds. Kluwere Academic Publishers, 1994.

[6] Suresh Singh, Mike Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom'98)*, 1998.

[7] R. Kravets, K. Schwan, and K. Calvert, "Power-aware communication for mobile computers," in *International Workshop on Mobile Multimedia Communications(MoMuc'99)*, 1999.

[8] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM 2000*, Apr. 2000.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, New York, 1979.

[10] M. Gerla L. Fratta and L. Kleinrock, "The flow deviation method: an approach to store-and-forward communication network design," *Networks*, vol. 3, pp. 97–133, 1973.

[11] D. Bertsekas and R. Gallager, *Data Networks (Second edition)*, Prentice Hall, 1992.

[12] J. L. Wu Y. J. Chang and H. J. Hu, "Optimal virtual circuit routing in computer networks," *IEE Proceedings I: Communications, Speech and Vision*, vol. 139, no. 6, Dec. 1992.

[13] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.

[14] "http://www.cc.gatech.edu/fac/ellen.zegura/graphs.html," .

[15] R. E. Newman-Wolfe and B. R. Venkatraman, "Performance analysis of a method for high level prevention of traffic analysis," in *Eighth Annual Computer Security and Applications Conference*, Nov. 1992.

[16] B. R. Venkatraman and R. E. Newman-Wolfe, "Transmission schedules to prevent traffic analysis," in *Ninth Annual Computer Security and Applications Conference*, Dec. 1993.

[17] B. R. Venkatraman and R. E. Newman-Wolfe, "Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network," in *Tenth Annual Computer Security and Applications Conference*, Dec. 1994.

[18] B. Radosavljevic and B. Hajek, "Hiding traffic flow in communication networks," in *MilCom'92*, Oct. 1992.

[19] Shu Jiang, Nitin H. Vaidya, and Wei Zhao, "Routing in packet radio networks to prevent traffic analysis," in *IEEE Information Assurance and Security Workshop*, June 2000.

[20] B. Gavish and S. L. Hantler, "An algorithm for optimal route selection in sna networks," *IEEE Trans. Commun., COM-31*, pp. 1154–1161, 1983.

[21] Frank Y. S. Lin and Jonathan Wang, "Minimax open shortest path first routing algorithms in networks supporting the smds service," in *International Conference on Communications(ICC'93)*, May 1993.

[22] Xin Yuan, "On the extended bellman-ford algorithm to solve two-constrained quality of service routing problems," in *International Conference on Computer Communications and Networks(ICCN'99)*, Oct. 1999.

[23] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," in *International Conference on Communications(ICC'98)*, June 1998.

[24] S. Chen and K. Nahrstedt, "An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions," *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, pp. 64–79, Nov. 1998.

[25] Q. Ma and P. Steenkiste, "Quality-of-service routing for traffic with performance guarantees," in *International Workshop on Quality of Service(IWQoS'97)*, May 1997.

[26] Liang Guo and Ibrahim Matta, "Search space reduction in qos routing," in *International Conference on Distributed Computing Systems*, May 1999.

[27] R. A. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions," in *GLOBECOM'97*, Nov. 1997.

Appendix

A. Proof of Theorem 1

We prove by induction. Assuming that before $k-$th iteration, $PATH(i)$ contains the optimal paths from node s to node i (through different input links at node i) with no more than $k-1$ hops, we prove that after $k-$th iteration, $PATH(i)$ contains the optimal paths with no more than k hops. From the algorithm, the path $p_{si}(j)$ in $PATH(i)$ is found by calculating the weights of all paths $p_{sj}(u)+(j,i)$ where $p_{sj}(u)$ is a path in $PATH(j)$ and choosing the one with minimal weight. We now prove two facts.

Fact 1: $p_{si}(j)$ is optimal. Let q_{sj} be any a path from s to j. Assume that u is the last node before j on path q_{sj}(See Fig. 7). Then the weight of path $q_{sj}+(j,i)$ can be computed by using the formula (26). If we denote c_1 as $\frac{e_{j,w}+\tau\Delta\lambda^C_{ji},+\rho\Delta\lambda^C_{uj}}{E_j}$ and c_2 as $\frac{e_{i,w}+\rho\Delta\lambda^C_{ji}}{E_i}$ for clarity, then we have

$$Weight(q_{sj}+(j,i)) = \max(Weight(q_{sj}), c_1, c_2). \quad (27)$$

As we know, the weight of path $p_{sj}(u)+(u,v)$ has similar form:

$$Weight(p_{sj}(u)+(u,v)) = \max(Weight(p_{sj}(u)), c_1, c_2) \quad (28)$$

Since $Weight(q_{sj}) \geq Weight(p_{sj}(u))$, then the following inequalities must be true.

$$\begin{aligned} Weight(q_{sj}+(j,i)) &\geq Weight(p_{sj}(u)+(u,v)) \quad (29)\\ &\geq Weight(p_{si}(j)). \quad (30)\end{aligned}$$

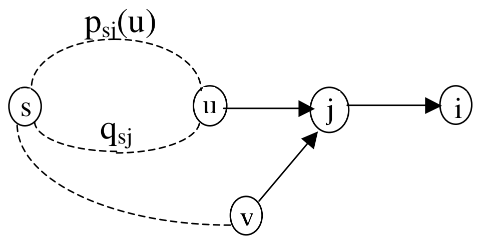


Fig. 7. Proof of Theorem 1

Fact 2: $p_{si}(j)$ contains no more than k hops. The portion of the path from s to j contains no more than $k-1$ hops. The last link (j, i) contributes one hop. So the number of hops in $p_{si}(j)$ cannot be more than k.

After $|N|-1$ iterations, $PATH(d)$ contains the optimal paths from s to d through different input links and each contains no more than $|N|-1$ hops. Since there are no negative links, the optimal path cannot have more than $|N|-1$ hops. In other words, the path in $PATH(d)$ with minimal weight is just the optimal path from s to d. This concludes the proof of the theorem. □