Preserving Traffic Privacy in Wireless Mesh Networks

Taojun Wu, Yi Cui and Yuan Xue
Department of Electrical Engineering and Computer Science
Vanderbilt University
Email: {*taojun.wu, yi.cui, yuan.xue*}*@vanderbilt.edu*

Abstract

Multi-hop wireless mesh network (WMN) has attracted increasing attention and deployment as a low-cost approach to provide last-mile broadband Internet access. Privacy is a critical issue in WMN, as traffic of an end user is relayed via multiple wireless mesh routers. Due to the unique characteristics of WMN, the existing solutions applied in Internet are either ineffective at preserving privacy of WMN users, or will cause severe performance degradation.

In this paper, we propose a light-weight privacy preserving solution aimed to achieve well-maintained balance between network performance and traffic privacy preservation. At the center of this solution is a novel metric called "traffic entropy", which quantifies the amount of information required to describe the traffic pattern and is used to characterize the performance of traffic privacy preservation. We further present a penalty-based shortest path routing algorithm that maximally preserves traffic privacy by minimizing the mutual information of "traffic entropy" observed at each individual relaying node, meanwhile controlling performance degradation within the acceptable region. Extensive simulation study proves the soundness of our solution.

1 Introduction

Recently, multi-hop wireless mesh network (WMN) has attracted increasing attention and deployment as a low-cost approach to provide last-mile broadband Internet access [1, 3, 4, 2, 12, 6]. In a WMN, each client accesses a stationary wireless mesh router. Multiple mesh routers communicate with one another to form a multi-hop wireless backbone that forwards user traffic to a few gateways connected to the Internet. Some perceived benefits of WMN include enhanced resilience against node failures and channel errors, high data rates, and low costs in deployment and maintenance.

Existing research on WMN has focused on how to better utilize the wireless channel resource and enhance its performance. Proposed solutions include equipping mesh routers with multiple radios and distributing the wireless backbone traffic over different wireless channels, routing the traffic through different paths [9, 20], or a joint solution of these two [17, 16]. Theoretical study shows that these approaches can significantly increase the capacity of WMN [15, 14]. These results make a significant step towards enabling WMN as an attractive alternative for broadband Internet access.

However, to further widen the deployment of WMN, and enable it as a competitive player in the market of broadband Internet access, privacy issue must be addressed. Privacy has been a major concern of Internet users [7]. It is a particularly critical issue in the context of WMN-based Internet access, where users' traffic is forwarded via multiple mesh routers. In a community mesh network, this means that the traffic of a residence can be observed by the mesh routers residing at its neighbors. Despite the necessity, limited research has been conducted towards privacy preservation in WMN.

This motivates us to investigate the privacy preserving mechanism in WMN. Two privacy issues are considered – data confidentiality and traffic confidentiality.

- *Data confidentiality*. It is obvious that data content reveals user privacy on what is communicated. Data confidentiality aims to protect the data content and prevent eavesdropping by intermediate mesh routers. Message encryption is a conventional approach for data confidentiality.

- *Traffic confidentiality*. Traffic information such as who the users are communicating with, when and how frequently they communicate, the amount and the pattern of traffic, also reveals critical privacy information. In a WMN, attackers can acquire such information via traffic analysis at mesh routers. While data confidentiality can be achieved via message encryption, it is much harder to preserve traffic confidentiality.

In this paper we focus on the user traffic privacy issue, and study the problem of traffic pattern concealment. In the existing literature, anonymous overlay routing [21, 5, 10, 13, 11, 8, 18] and traffic padding [19] have been proposed to preserve user traffic privacy and increase the difficulty for traffic analysis. The former approach provides user anonymity in an end-to-end connection through layered encryption and multi-hop overlay routing. The latter one conceals the traffic shape by generating a continuous random data stream at the link level. However neither of them can be applied to WMN directly. First, the number of nodes in a WMN is limited. Second, traffic forwarding relationship among nodes is strongly dependent on their locations and the network topology. To better utilize the wireless channel resource and enhance the data delivery performance, a short path is usually selected; or a load-balanced routing scheme is employed. Such observations show that the anonymity systems, which rely on relaying traffic among nodes (randomly selected out of thousands) to gain anonymity, can not effectively preserve users' privacy in WMN, or at the cost of significant performance degradation. On the other hand, traffic padding mechanism consumes a considerable amount of network bandwidth, which makes it impractical in resource-constrained WMNs.

In light of these problems, we aim at designing a light-weight privacy preserving mechanism for WMN which is able to balance the traffic analysis resistance and the bandwidth cost. Our mechanism makes use of the intrinsic redundancy of WMN, which is able to provide multiple paths for data delivery. By intuition, if the traffic from the source (*i.e.*, gateway) to the destination (*i.e.*, mesh router) is split to many paths, then all the relaying nodes[1] along the paths could only observe a portion of the entire traffic. Moreover, if the traffic is split in a random way both spatially and temporally, then an intermediate node has limited knowledge to figure out the overall traffic pattern. Thus the traffic pattern is concealed.

Based on this intuition, we seek a routing scheme such that the statistical distributions of the traffic observed at intermediate relaying nodes are independent from the actual traffic from the source to the destination. To achieve this goal, we first define an information-theoretic metric – "*traffic entropy*", which quantifies the amount of information required to describe the traffic pattern. Then we present a penalty-based routing algorithm, which aims to minimize the mutual information of "traffic entropy" observed at each relaying node, meanwhile controlling the network performance degradation under the acceptable level.

The main contributions of this paper are as follows. First, to the best of our knowledge, it is the first work that identifies the privacy issue in WMN and presents a light-weighted privacy preserving architecture for WMN. Second, "traffic entropy" provides a unified metric, based on which performance of all traffic privacy preserving solutions can be quantitatively evaluated and fairly compared. Finally, different to traditional solutions applied to Internet, our algorithm strives to achieve tunable balance between privacy preservation and routing performance, making it suitable for resource-restrained WMNs.

The rest of this paper is organized as follows. In Section 2, we present the overall architecture for privacy preservation in WMN. Section 3 and 4 focus on the traffic privacy issue. In particular, Section 3 presents the model to quantify the performance of traffic privacy preservation, and Section 4 presents the routing algorithm. The proposed privacy preserving solution is evaluated via extensive simulation study in Section 5. Section 6 concludes the paper and points out the future directions.

[1] In this paper, we use the following terms interchangeably: wireless mesh router, intermediate relaying node, wireless node.

2 Privacy Preserving Architecture for Wireless Mesh Network

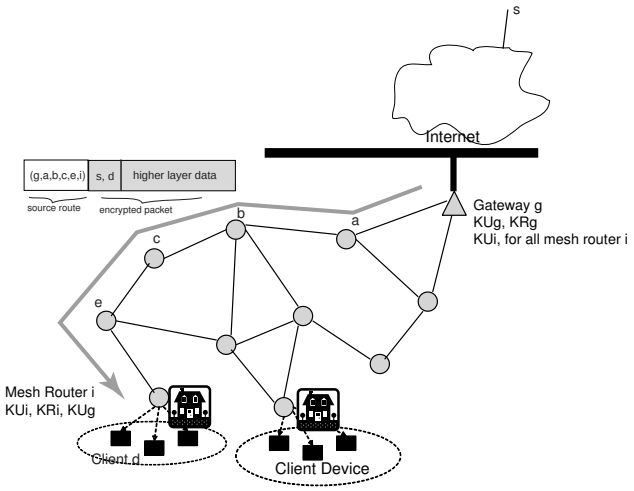


Figure 1. Privacy preserving architecture for wireless mesh network.

We consider a multi-hop WMN shown in Fig. 1. In this mesh network, client devices access a stationary wireless mesh router at its residence. Multiple mesh routers communicate with one another to form a multi-hop wireless backbone that forwards user traffic to the gateway which is connected to the Internet.

Two privacy aspects are considered. *Data confidentiality* aims to protect the data content from eavesdropping by the intermediate mesh routers. *Traffic confidentiality* prevents the traffic analysis attack from the mesh routers, which aims at deducing the traffic information such as who the user is communicating with, the amount and the pattern of traffic. Our privacy preserving architecture aims to protect the privacy of each wireless mesh router, the basic routing unit in WMN. The architecture consists of the following functional components.

- *Key Distribution*. In this architecture, each mesh node, as well as the gateway, has a pair of public and private keys (KU, KR). The gateway maintains a directory of certified public keys of all mesh nodes. And each mesh node has a copy of the public key of the gateway KU_g. The public key KU_i of mesh node i and KU_g are used to establish the shared secret session key KS_{gi}, which is used to encrypt the messages between them.

- *Message Encryption*. Let M be the IP packet sent from a source s in the Internet to a client d in the mesh network, and i be the mesh router of client d. The whole IP packet M, which contains the original source and destination address s and d, is encrypted at gateway g via the shared secret key KS_{gi}: $M_e = E(KS_{gi}, M)$. To route the encrypted packet M_e to its destination, the gateway prefixes the source route from the gateway g to the router i to the packet. The encapsulated packet is then forwarded by relaying routers in WMN. Likewise, packets traveled in the reversed direction are treated in the same way. As the source address s and other higher layer header information, such as port, are all encrypted, the relaying routers are unable to obtain the information on who the client of router i is communicating with, and what type of application is involved. Since encryption and decryption take place only at the gateway and the destination mesh router, much less computation is required, which is a desired feature in WMN.

- *Routing Control*. With source route in clear text in an encapsulated packet, the intermediate mesh routers can still observe the amount and the pattern of the traffic of a particular mesh node i. To address this problem, our privacy preserving mechanism explores the path diversity of WMN, and forwards packets between the gateway and the mesh node via different routes. Thus any relaying router can only observe a portion of the whole traffic of this connection. In Section 4, we detail the design of a penalty-based routing algorithm, which randomly selects a route for each individual packet such that the observed traffic pattern at each relaying node is independent of the overall traffic. In our design, the gateway maintains a complete topology of the WMN, and computes the source routes between the destination mesh nodes and itself.

3 Model

3.1 Network Model

We model the WMN shown in Fig. 1 as a graph $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of wireless nodes in WMN, and $\mathcal{E}$ is the set of wireless edges (x, y) between any two nodes x, y. Each node x maintains a logical connection with the gateway node g. Node x receives data with the Internet via g. The source and destination information of a packet is open to the relaying node. The traffic pattern of x can be categorized into two types: incoming traffic pattern and outgoing traffic pattern. In this paper, we mainly consider the first type.

If the traffic between s and x goes through only one route, then any relaying node on this route can easily observe the entire traffic between g and x, thus violating its traffic pattern privacy. To avoid this problem, x must establish multiple paths with g and distribute its traffic along these paths, such that any node can only get partial picture of x's traffic pattern.

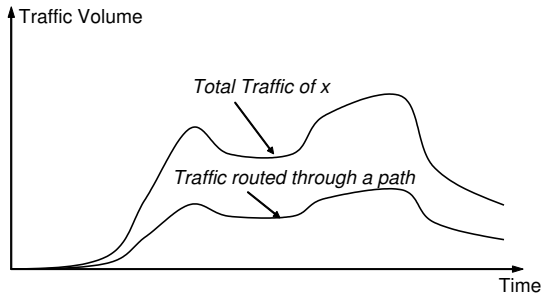


Figure 2. An Example of Isomorphic Traffic

However, the complete traffic pattern information of x could still be obtained by a single node in case of multi-path routing. In the example shown by Fig. 2, g allocates the traffic to x via three disjoint routes by fixed proportion. Then for any node along any path, although only seeing one third of the flow, the observed traffic shape is isomorphic to the original one. Therefore, the traffic to x must be distributed along multiple route in a time-variant fashion, such that the traffic pattern observed at any node is statistically deviant from the original pattern.

3.2 Traffic Entropy

We propose to use information entropy as the metric to quantify the performance of a solution at preserving the traffic pattern confidentiality. In what follows, we consider two nodes x and y. x is the destination node of the traffic from the gateway g to x. y is the observing node, which relays packets for x and also tries to analyze the traffic of x.

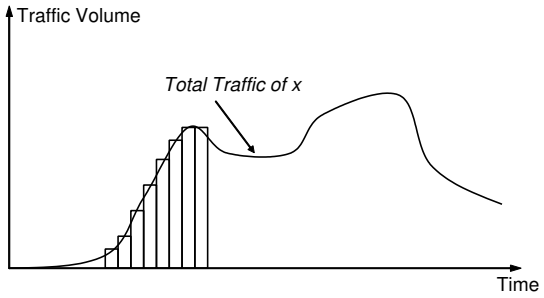


Figure 3. Sampling-based Traffic Analysis

$\mathcal{V}$	wireless node set
$\mathcal{E}$	edge set
g	gateway node
x	destination node
y	observing node
X	random variable describing x's traffic pattern
Y^X	random variable describing x's traffic pattern observed by y
$H(X)$	entropy of X
$H(Y^X)$	entropy of Y^X
$I(Y^X, X)$	mutual information between X and Y^X

Table 1. Notations used in Sec. 3

3.2.1 Basic Definition

Ideally, we view the traffic of x as a continuous function of time, as shown in Fig. 3. In practice, the traffic analysis is conducted by dividing time into equal-sized sampling periods, then measuring the amount of traffic in each period, usually in terms of number of packets, assuming the packet sizes are all equal. Therefore, as the first step, we discretize the continuous traffic curve into piece-wise approximation of discrete values, each

denoting the number of packets destined to x in a sampling period.

Now, we use X as the random variable of this discrete value. Y^X is the random variable representing the number of packets destined to x observed at node y in a sampling period. We denote $P(X = i)$ as the probability that the random variable X is equal to i ($i \in \mathcal{N}$), i.e., the probability that node x receives i packets in a sampling period. Likewise, $P(Y^X = j)$ is the probability that Y^X is equal to j ($j \in \mathcal{R}$), i.e., j packets destined to x go through node y in a sampling period.

Then the discrete Shannon entropy of the discrete random variable X is

$$H(X) = -\sum_i P(X = i) \log_2 P(X = i) \quad (1)$$

$H(X)$ is a measurement of the uncertainty about outcome of X. In other words, it measures the information of node x's traffic, *i.e.*, the number of bits required to code the values of X. $H(X)$ takes its maximum value when the value of X is uniformly distributed. On the other hand, if the traffic pattern is CBR, then $H(X) = 0$ since the number of packets at any sampling period is fixed[2].

Similarly, we have the entropy for Y^X as follows.

$$H(Y^X) = -\sum_j P(Y^X = j) \log_2 P(Y^X = j) \quad (2)$$

3.2.2 Mutual Information

We then define the conditional entropy of random variable Y^X with respect to X as

$$H(X|Y^X) = -\sum_j P(Y^X = j) \sum_i p_{ij} \log_2 \ p_{ij} \quad (3)$$

where $p_{ij} = P(X = i|Y^X = j)$ is the probability that $X = i$ given condition that $Y^X = j$. $H(X|Y^X)$ can be thought of as the uncertainty remaining about X after Y^X is known. The joint entropy of X and Y^X can be shown as

$$H(X, Y^X) = H(Y^X) + H(X|Y^X) \quad (4)$$

[2]This offers the information-theoretic interpretation for traffic padding: by flattening the traffic curve with blank packets, the entropy of observable traffic is reduced to 0, which perfectly hides the information of the original traffic pattern.

Finally, we define the mutual information between X and Y^X as

$$\begin{aligned} I(Y^X, X) &= H(X) + H(Y^X) - H(X, Y^X) \\ &= H(X) - H(X|Y^X) \end{aligned} \quad (5)$$

which represents the information we gain about X from Y^X.

Back to the example in Fig. 2, let us assume that the observing node y is located on one route destined to x. Since the traffic shape observed at y is the same as x, at any sampling period, if $Y^X = j$, then X must equal to a fixed value i, making $P(X = i|Y^X = j) = 1$. According to Eq. (3), this makes the conditional entropy $H(X|Y^X) = 0$. According to Eq. (5), we have $I(Y^X, X) = H(X)$, implying that from Y^X, we gain the complete information about X.

On the contrary, if Y^X is independent from X, then the conditional probability $P(X = i|Y^X = j) = P(X = i)$, which maximizes the conditional entropy $H(X|Y^X)$ to $H(X)$. According to Eq. (5), we have $I(Y^X, X) = 0$,[3] i.e., we gain no information about X from Y^X.

In reality, since Y^X records the number of a subset of packets destined to node x, it can not be totally independent from the random variable X. Therefore, the mutual information should be valued between the two extremes discussed above, *i.e.*, $0 < I(Y^X, X) < H(X)$. This means that node y can still obtain partial information of X's traffic pattern. However, a good routing solution should minimize such mutual information as much as possible for any potential observing node. More formally, we should minimize

$$\max_{Y \in \mathcal{V}-X} I(Y^X, X) \quad (6)$$

the maximum mutual information that any node can obtain about X.

4 Penalty-based Routing Algorithm

In this section, we propose a penalty-based routing algorithm to achieve our goal of hiding traffic pattern by exploiting the richness of available paths between two nodes in WMN. Specifically, we choose to adopt

[3]By the definition of mutual information, $I(Y^X, X) \geq 0$, with equality if and only if X and Y are independent.

the *source routing* scheme. Such a choice is enabled by the fact that one node can easily acquire the topology of the WMN it belongs to, which is mid-sized (within 100 nodes) and static.

When designing the algorithm, we also keep in mind the need to compromise between sufficient security assurance and acceptable system overhead. We would show in our algorithm that system performance is satisfactory and security assurance is adequate.

Shown in Tab. 2, the algorithm operates in three phases, *path pool generation*, *candidate path selection* and *individual packet routing*.

First, in the path pool generation phase, we try to generate a large set of diversified routing paths connecting the gateway g and the destination node x, denoted as S_{paths}. The path generation algorithm is an iterated process of applying a modified version of Dijkstra's algorithm. Here, each node is assigned a penalty weight, and the weight of an edge is defined as weighted average of penalty weights of its two end nodes. The weight (or cost) of a path is defined as the sum of penalty weights of all edges consisting this path. The algorithm runs in iterations. Initially, we set the penalty weight of each node as 1, then run the Dijkstra's algorithm to find the first shortest path from the gateway g to x. Next, we increase the penalty weight for each node on this found path. This will make these appeared nodes less competitive to other nodes in becoming components of next path. After this, the algorithm proceeds to the next iteration, generating the second path, and all nodes appearing on the second path are penalized through increasing their weights. This process goes on until enough number of paths are found.

Second, in the candidate path selection phase, we try to choose a combination of diversified routing paths, a subset of paths from the set S_{paths} from g to x, denoted as $S_{selected}$. The paths in $S_{selected}$ are selected randomly from S_{paths}. After each choice of a path into $S_{selected}$, the probability factor of that path is decreased to lower the chance of multiple identical paths existing in $S_{selected}$. $S_{selected}$ is changed and renewed corresponding to network activities.

Third, in the packet routing phase, we choose randomly from $S_{selected}$ one path for each packet and increase the counter for the selected path subset $S_{selected}$. This $S_{selected}$ path subset expires after counter reaches its predetermined threshold. Then $S_{selected}$ is renewed by calling the second phase again.

Since packets are assigned a randomly chosen path, and all these candidate paths are designed to be disjoint, the chance that packets are routed in similar paths is small. This is shown in our experiment results.

This algorithm is designed to balance the needs of routing performance (finding paths with smallest hop count) and preserving traffic pattern privacy (finding disjoint paths). The penalty weight update function serves as the tuning knob to maneuver the algorithm between these two contradictory goals. During the initialization, when the penalties of all nodes are equal, the path found by the algorithm is indeed shortest in terms of hop count. As a node is chosen by more routes, its penalty weight monotonically increases, making it less likely to be chosen again. Thus, as the algorithm proceeds, the newly-chosen paths (shortest in terms of its aggregate penalty weight) become more disjoint from existing paths, but longer in terms of hop count. The pace of such shift from "smallest hop-count path" to "disjoint path" is controlled by how fast the penalty weight update function grows. Our experiment results confirm us this reasoning. Finally, by randomly assigning packets along different paths, the algorithm maximally disturbs the traffic pattern of any $g - x$ pair.

5 Experimental Results

5.1 Simulation Setup

We base our simulation on a randomly generated topology (Fig. 4) (600 x 600) with 30 nodes. The effective distance between two nodes is set to be 250. The whole process of simulation consists of 400,000 logical ticks. In each single tick, a packet is generated at gateway node 0 and its destination is randomly decided to be one of the other 29 nodes. To better simulate real network traffic, we set the probability of 0.05 that at one tick no packet is generated, i.e., idle probability. The distance delay factor is chosen to be 0.003 tick and hop delay factor is decided as 0.05 tick. We approximate hop delay at any node by multiplying hop delay factor with its usage count by all paths chosen initially.

With a relatively small node set, we choose 50 as

/*Penalty-Based Shortest Path*
$PBSP(Snode, Dnode)$
  For each node $v \in \mathcal{V}$
    $d[v] \leftarrow \infty$
  For each node $v \in \mathcal{V}$
    $prev[v] \leftarrow \infty$
  For each node $v \in \mathcal{V}$
    $visited[v] \leftarrow 0$
  $d[SNode] \leftarrow 0$
  Repeat
    Get unvisited vertex v with the least $d[v]$
    If $d[v] \geq \infty$**, Then** v unreachable
    Else $visited[v] \leftarrow 1$
    For all v's neighbors w
      $EdgePenalty = \alpha[pow(\gamma, (w.tag))] + \beta(v.tag)$
      If $d[w] > d[v] + EdgePenalty$
        $d[w] \leftarrow d[v] + EdgePenalty$
        $prev[w] \leftarrow v$
  Until $visited[v] = 1, \forall v \in \mathcal{V}$

/*Generate S_{paths} For Each $g - x$ Pair*/
$GenPath()$
For All Non-Gateway Nodes x
  For each node $v \in \mathcal{V}$
    $v.tag \leftarrow 1$
  Repeat
    PBSP(g, x)
    Get new $g - x$ path P_{new} from vector $prev[]$
    Store P_{new} in S_{paths}
    For all nodes v on P_{new}
      $v.tag \leftarrow v.tag + 1$
  Until *PathPoolSize* paths found.

/*Select $S_{selected}$ For Each $g - x$ Pair*/
$SelPath()$
Repeat
  $rnd = rand()\ mod\ PathPoolSize$
  select rndth path from S_{paths}
Until $SelPathNum$ paths selected

/*Decide path for arriving packet*/
$RoutePkt(Snode, Dnode)$
  $Packets[Dnode] \leftarrow Packets[Dnode] + 1$
  $rndpath = rand()\ mod\ SelPathNum$
  route packet along the $rndpath$th path from $S_{selected}$
  If $Packets[Dnode] > ReSelPathCnt$
    $Packets[Dnode] \leftarrow 0$
    SelPath()

Table 2. Penalty-based Routing Algorithm

v, w	node
$v.tag$	number of times v is included by a path
α	factor to slow down penalty rate
β	factor to avoid many identical paths in beginning stages of path generation
γ	base of exponential penalty function
$d[]$	penalty vector for every node
$prev[]$	vector to store P_{new} reversely
$Packets[]$	vector to store number of arrived packets for every node

Table 3. Notations used in Sec. 4

our $PathPoolSize$ and 5 as $SelPathNum$. The selected path subset $S_{selected}$ for any destination node is renewed after sending 50 packets to that node. To obtain multiple diversified paths with Dijkstra's algorithm more quickly, we introduce exponential penalty function on tag of one node and used γ as the base of exponential function when deciding on which edge to include to candidate path. To slow down growing rate of exponential penalty function, we multiply the exponential function with a factor α when calculating $EdgePenalty$. To avoid getting too many identically paths in beginning stages, we amplify influence of another node by multiplying tag of another node with β. The penalty parameters α, β, γ are chosen to be 0.5, 15 and 1.85, respectively.

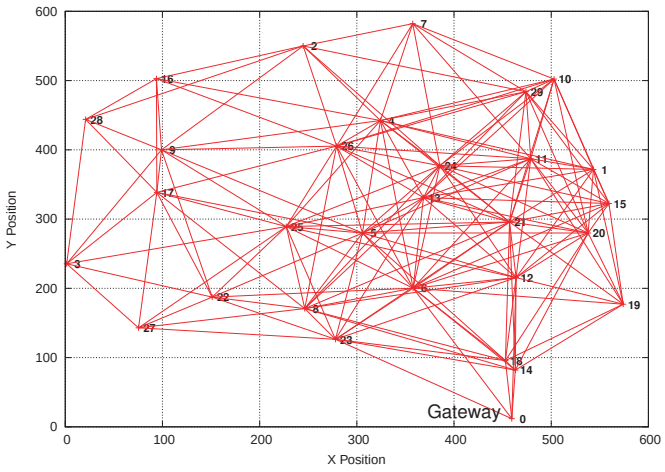
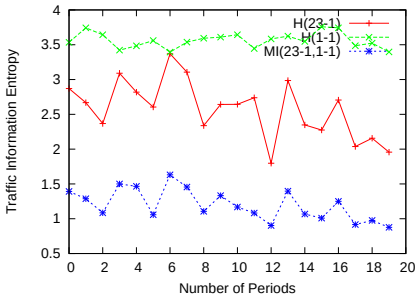


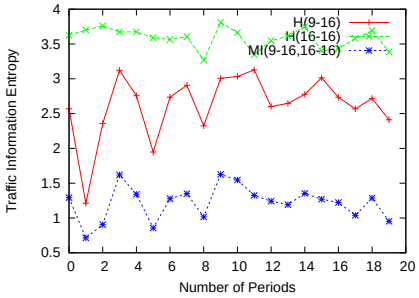
Figure 4. Experimental Topology

5.2 Traffic Entropy and Mutual Information

The total 400,000 ticks is divided into 20 periods. Each period is then divided into 50 intervals and one interval is 400 ticks long. Within each interval, for each destination node x, we count the number of packets that all other nodes y has relayed for x. Then for each period, we independently calculate the traffic entropies $H(X)$, $H(Y^X)$, and mutual information $I(Y^X, X)$ based on their definitions in Sec. 3.2.



(a) Destination: Node 1, Observer: Node 23

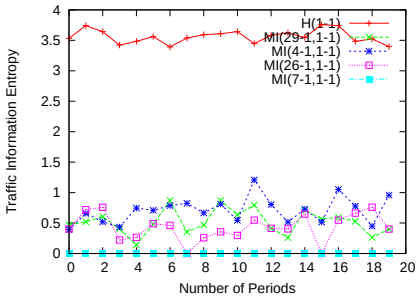


(b) Destination: Node 16, Observer: Node 9

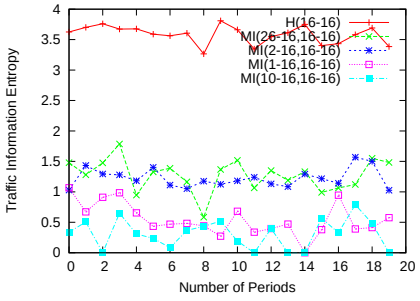
Figure 5. Traffic Entropy along Time (Single Observer, $\gamma = 1.85$)

Due to the space limit, we only show part of our results. Among all nodes in the network, we choose two sets of nodes. Nodes in the first set $\{1, 6, 11, 15, 23, 24, 25, 29\}$ are close to (2 to 3 hops) the gateway node 0. Nodes in the second set $\{2, 3, 7, 16, 17, 28\}$ are at the edge of the network, 4 to 5 hops away from the gateway. We choose two representative nodes, 1 and 16, out of each set.

Fig. 5 shows the variance of traffic entropy and mutual information along the time. In Fig. 5 (a), $H(1-1)$ denotes the traffic entropy of node 1. $H(23-1)$ denotes the traffic entropy of node 23 based on its observation on node 1. $MI(23-1, 1-1)$ denotes the mutual information node 23 shares with node 1. The same notation rules apply for Fig. 5 (b), where node 16 is the destination, and 9 is the observer. In both pictures, the observing node only shares 40% or less of information about the observed destination node at any sampling period.



(a) Destination: Node 1, Observers: Node 4, 7, 26, and 29



(b) Destination: Node 16, Observers: Node 1, 2, 10, and 26

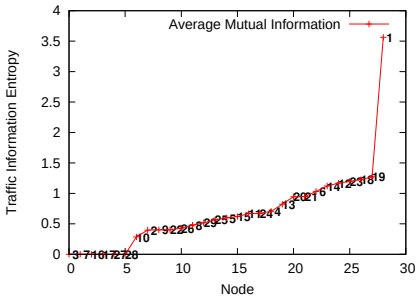
Figure 6. Traffic Entropy in Different Sampling Periods (Multiple Observers, $\gamma = 1.85$)

This observation is further confirmed in Fig. 6, where we plot the time-variant mutual information that destinations 1 and 16 share with other randomly-chosen observing nodes. These results show that with our algorithm, the destination node is able to consistently limit the proportion of mutual information it shares with the observing nodes.

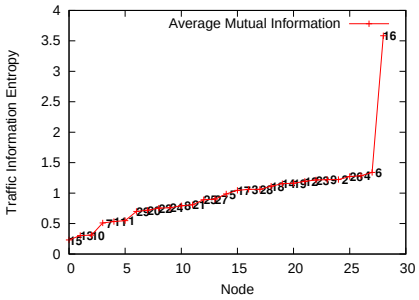
5.3 Which Nodes have more Mutual Information?

In Fig. 7 (a), we calculate the time-averaged mutual information for all observing nodes with respect to the destination node 1, and sort them in the ascending order. Here, we observe an almost linearly-growing curve except at its head and tail. For nodes at the head of the cure, their mutual information is 0 since they lie at the outer rim of the network, hence are not cho-

sen by our routing algorithm to relay traffic for node 1. At the tail of the curve is destination node 1, whose mutual information is actually the traffic entropy of its own. In Fig. 7 (b), we observe the same phenomenon for destination 16, except at the head of the curve. This is because its network location is at the opposite end of the gateway, making every node of the network to be its candidate relaying node.



(a) Destination: Node 1 ($\gamma = 1.85$)

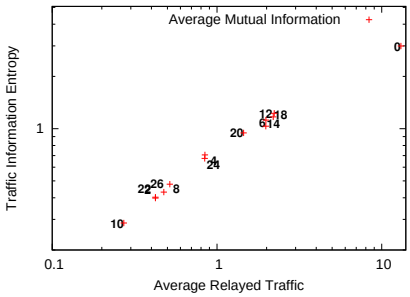


(b) Destination: Node 16 ($\gamma = 1.85$)

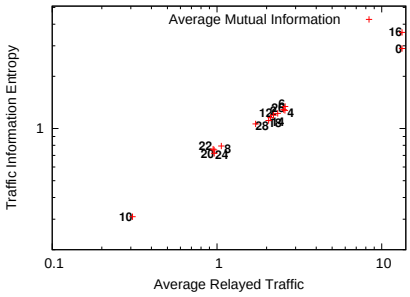
Figure 7. Sorted Mutual Information

This leads us to investigate if such distribution of mutual information is related with any other factors. We tried to connect mutual information of each node with certain metric, such as its distance to the destination, but failed to find any causal relationship. We then sort observing nodes based on the averaged relayed traffic (average number of packets each node relays in a sampling period) on a log-log scale, and find the linear distribution as shown in Fig. 8.

Obviously, such a power-law correlation tells us that more traffic an observing node relays for a destination node, the more mutual information can be obtained about its traffic entropy. Furthermore, it gives us one way to experimentally quantify the relationship of these two metrics. Let T be the amount of traffic relayed and I be the mutual information, then their



(a) Destination: Node 1 ($\gamma = 1.85$)



(b) Destination: Node 16 ($\gamma = 1.85$)

Figure 8. Power-law Correlation of Mutual Information and Amount of Traffic Relayed

power-law relationship can be written as

$$I = aT^k \tag{7}$$

where a is the constant of proportionality and k is the exponent of the power law, both of which can be measured from Fig. 8. If $k < 1$, then the mutual information of an observing node grows in a sub-linear fashion as the amount of its relayed traffic increases, and in a super-linear fashion otherwise. From what we have in Fig. 8 and the same results for other destination nodes, $k < 1$. This means that each time to make its mutual information further grows with the same increment, an observing node has to relay more and more traffic.

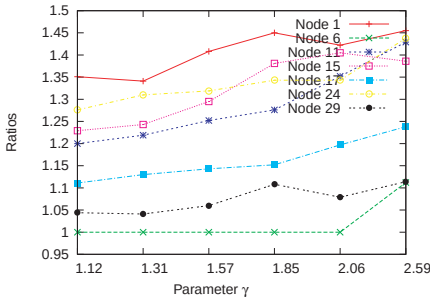
5.4 Tradeoff between Performance Degradation and Traffic Privacy

Finally, we study the performance tradeoff of our algorithm by tuning its exponential penalty function base γ. The performance degradation introduced by our algorithm is captured by the average hop ratio. For each gateway-destination pair $g - x$, this metric is defined as the ratio between the average number of hops a

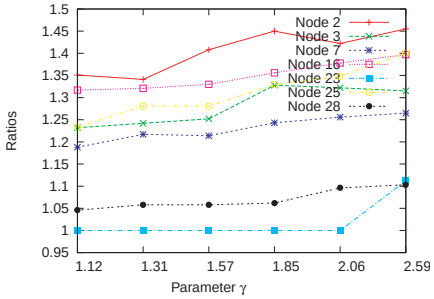
packet goes through using our algorithm and the number of hops of the shortest path between g and s. From Fig. 9, we can see that the average hop ratio increases as γ increases. The direct neighbors of the gateway are less sensitive to the change of γ, like node 6 in Fig. 9(a) and node 23 in Fig. 9(b).

In Fig. 10 and Fig. 11 we find that under shortest path routing, the mutual information of a node is 0 if it is not on the path to destination node. Otherwise, the mutual information node is much higher than the case of our algorithm. Also worth noting is that increasing of γ has different impact on different node, depending on its distance to gateway, destination, and its location in the WMN. Take node 6 (Fig. 11) and 12 (Fig. 10) for example, since they lie near to gateway node and are relatively centrally situated, their observed mutual information vary little with respect to the change of γ. Whereas for node 17 (Fig. 10), which is far away from destination node 1 and on edge of WMN, mutual information shared between itself and node 1 increases with the growth of γ, indicating more traffic is routed through farther nodes. This tendency of routing packets from farther nodes leads to higher average number of hops, which is confirmed by our analysis about average hop ratio. The great fluctuation of node 26 is due to its position in center of topology and equal distance to both gateway and destination.

We also observe from Fig. 12 that our algorithm achieves our goal of preserving traffic pattern. In the first place, it is easy to conclude that in normal shortest path routing, all relaying nodes shares the same traffic information with destination node, as shown by the tail of the ShortestPath curve in Fig. 12. However, for our algorithm, the mutual information shared between relaying nodes and destination node varies much less among all relaying nodes. And the higher γ is, the more leveled off the curve becomes, and the closer we are to the goal of minimizing the greatest mutual information, formulated in Eq. 6. It is also interesting to observe that mutual information is 0 for some nodes far away from both gateway and destination. For example, in Fig. 12 (a), when destination is 1, while all nodes participate in relaying packets for destination 16, since destination and gateway nodes are in opposite directions with respect to WMN topology.

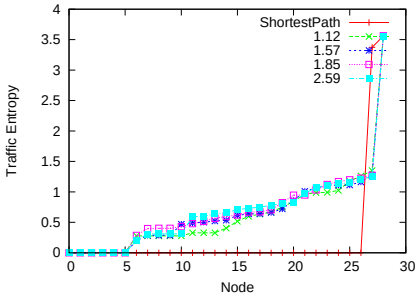


(a) Hop Ratio of Nodes in the First Set

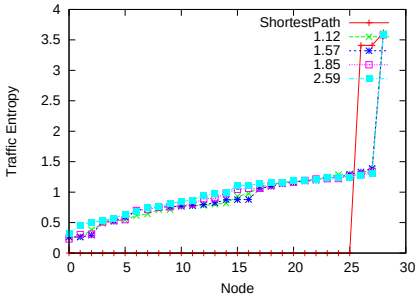


(b) Hop Ratio of Nodes in the Second set

Figure 9. Average Hop Ratio

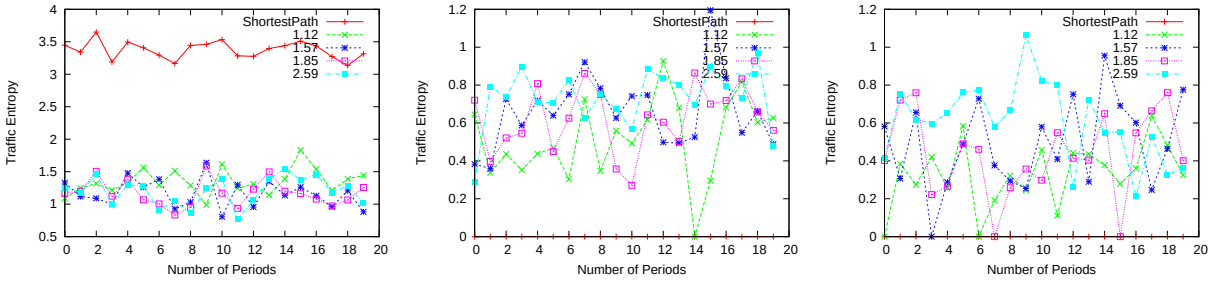


(a) Destination: Node 1



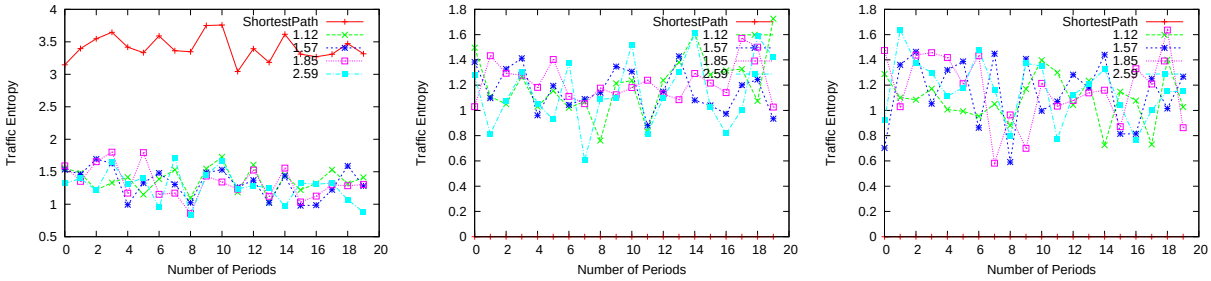
(b) Destination: Node 16

Figure 12. Sorted Mutual Information under Different Penalty Parameters



(a)Observer: Node 12 (b) Observer: Node 17 (c) Observer: Node 26

Figure 10. Mutual Information under Different Penalty Parameters (Destination: Node 1)



(a)Observer: Node 6 (b) Observer: Node 2 (c) Observer: Node 19

Figure 11. Mutual Information under Different Penalty Parameters (Destination: Node 16)

6 Conclusion

This paper identifies the problem of traffic privacy preservation in wireless mesh networks (WMN). To attack this problem, we start by introducing a light-weight architecture for WMN, then propose "traffic entropy", an information theoretic metric to quantify how well a solution performs at preserving the traffic pattern confidentiality, all of which pave the way to our penalty-based shortest path routing algorithm. Simulation results show that our algorithm is able to maximally preserve the traffic privacy, meanwhile managing the network performance degradation within the acceptable region.

For the future work, we will focus on the following problems. First, multiple observing nodes may collude to analyze the traffic pattern of a destination node. Besides new routing solutions to defend collusion, we also need to extend the "traffic entropy" concept by applying the chain rules in information theory. Second, although our algorithm is evaluated in a single-radio, single-channel WMN setting, it can be easily enhanced to exploit the advantage of multiple radios and multiple channels available in WMNs. Performance evaluation of the enhanced algorithm in such settings will be an interesting future work.

References

[1] Mesh networks inc. `http://www.meshnetworks.com`.

[2] Mit roofnet. `http://www.pdos.lcs.mit.edu/roofnet/`.

[3] Radiant networks. `http://www.radiantnetworks.com`.

[4] Seattle wireless. `http://www.seattlewireless.net`.

[5] A. Back, U. Möller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Information Hiding Workshop (IH)*, 2001.

[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM MOBICOM*, pages 31–42, 2005.

[7] R. Clarke. Internet privacy concerns confirm the case for intervention. *Communications of the ACM*, 42(2):60–67, 1999.

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The wimeshd-generation onion router. In *USENIX Security Symposium*, 2004.

[9] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MOBICOM*, pages 114–128. ACM Press, 2004.

[10] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *ACM Conference on Computer and Communications Security (CCS)*, 2002.

[11] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):39–41, 1999.

[12] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: The case for taps. In *HotNets*, 2003.

[13] S. Katti, , D. Katabi, and K. Puchala. Slicing the onion: Anonymous routing without pki. Technical report, MIT CSAIL Technical Report 1000, 2005.

[14] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *ACM MOBICOM*, 2005.

[15] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *ACM MOBICOM*, pages 43–57, New York, NY, USA, 2005.

[16] A. Raniwala and T. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE INFOCOM*, 2005.

[17] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *Mobile Computing and Communications Review*, 8(2):50–65, 2004.

[18] M. G. Reed, P. F. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[19] W. Stallings. *Cryptography and Network Security*. Prentice Hall, 2003.

[20] Y. Yuan, H. Yang, S. H. Y. Wong, S. Lu, and W. Arbaugh. Romer: Resilient opportunistic mesh routing for wireless mesh networks. In *Proc. of IEEE WiMesh*, 2005.

[21] L. Zhuang, F. Zhou, B. Y. Zhao, and A. Rowstron. Cashmere: Resilient anonymous routing. In *Symposium on Networked Systems Design and Implementation (NSDI)*, 2005.