Secure Data Communication in Mobile Ad Hoc Networks

Panagiotis Papadimitratos, *Member, IEEE,* and Zygmunt J. Haas, *Senior Member, IEEE*

***Abstract*—We address the problem of secure and fault-tolerant communication in the presence of adversaries across a multihop wireless network with frequently changing topology. To effectively cope with arbitrary malicious disruption of data transmissions, we propose and evaluate the secure message transmission (SMT) protocol and its alternative, the secure single-path (SSP) protocol. Among the salient features of SMT and SSP is their ability to operate solely in an end-to-end manner and without restrictive assumptions on the network trust and security associations. As a result, the protocols are applicable to a wide range of network architectures. We demonstrate that highly reliable communication can be sustained with small delay and small delay variability, even when a substantial portion of the network nodes systematically or intermittently disrupt communication. SMT and SSP robustly detect transmission failures and continuously configure their operation to avoid and tolerate data loss, and to ensure the availability of communication. This is achieved at the expense of moderate transmission and routing overhead, which can be traded off for delay. Overall, the ability of the protocols to mitigate both malicious and benign faults allows fast and reliable data transport even in highly adverse network environments.**

***Index Terms*—Fault tolerance, mobile ad hoc network (MANET) security, multipath routing, network security, secure data transmission, secure message transmission, secure routing.**

I. Introduction

THE EMERGING technology of *mobile ad hoc networking* (MANET) is based on wireless multihop architecture without fixed infrastructure and prior configuration of the network nodes. The salient features of this new networking paradigm include: 1) collaborative support of basic networking functions, such as routing and data transmission; 2) lack of administrative boundaries of the network nodes; 3) absence of a central entity in the network; and 4) transient, in general, associations of the network nodes. As a result, a node cannot make any assumption about the trustworthiness of its peers, which assist the node with its communication and, in general, does not possess their credentials.

Securing the basic network operation becomes one of the primary concerns in ad hoc networks and, in fact, a prerequisite for reliable and quality-of-service (QoS) communication in adversarial environments. The challenge lies in securing communication and maintaining connectivity in the presence of adversaries, across an unknown, frequently changing multihop wireless network topology. To address this complex problem and provide comprehensive security, both phases of the communication, the route discovery and the data transmission, must be safeguarded.

Recently, a number of works proposed secure routing mechanisms to defend against a range of attacks under different assumptions and system requirements [1]–[8]. However, secure routing protocols alone, which ensure the correctness of the route discovery, cannot guarantee secure and undisrupted delivery of data. In other words, a correct, up-to-date route cannot be considered automatically free of adversaries. An intelligent adversary can, for example, follow the rules of the route discovery, place itself on a route, and later start redirecting traffic, dropping, or forging and injecting data packets. Clearly, an adversary can hide its malicious behavior for a long period of time and strike at the least expected time. Thus, it is impossible to discover such an adversary prior to its attack.

MANET routing, as well as secure routing protocols assume mechanisms, such as reliable data link layer and route maintenance, which were not designed for and cannot cope with malicious disruptions of the data transmission. Reliable transport protocols cannot address the problem either: an attacker can forge, for example, transmission control protocol (TCP) acknowledgment, while dropping data packets, misleading two communicating nodes that the data flow is undisrupted. End-to-end security such as the IP-Security (IPSec) [9] *authentication header* (AH) protocol [10] can prevent adversaries from forging or corrupting data and feedback. But IPsec does not allow the sender to detect loss of data and, thus, take any corrective action. Nor the combination of security services and reliable transport [e.g., *stream control transmission protocol* (SCTP) [11]] provides an effective solution: a communication failure can be detected, but the same, structurally intact yet compromised path will be repeatedly utilized, because the transport layer protocol cannot influence the choice of the route in the network. Finally, multipath transmissions [12]–[14], can protect against failures. However, "blind" redundant transmissions alone can be highly inefficient without a robust mechanism to detect transmission failures and adapt to the network loss conditions.

Our contribution is a novel, general solution, tailored to the MANET requirements, to effectively and efficiently secure the data transmission phase: the *secure message transmission* (SMT) and *secure single-path* (SSP) protocols. We emphasize

Manuscript received October 12, 2004; revised August 15, 2005. This work was supported in part by the National Science Foundation (NSF) under Grant ANI-9980521, in part by the DoD MURI programs administered by the Office of Naval Research under Grant N00014-00-1-0564, and in part by the Air Force Office of Scientific Research under Grant F49620-02-1-0233.

P. Papadimitratos was with Cornell University, Ithaca, NY 14853 USA. He is now with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: papadp@vt.edu).

Z. J. Haas is with the Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: haas@ece.cornell.edu).

Digital Object Identifier 10.1109/JSAC.2005.861392

that the goal of SMT and SSP is not to securely discover routes in the network—they assume that secure discovery of routes has been already performed, although routes may not be free of adversaries. Then, the goal of SMT and SSP, whose basic ideas we presented in [15] and [16], is to secure the data transmission: SMT and SSP operate without restrictive assumptions on the network trust and security associations, promptly detect and avoid nonoperational or compromised routes, tolerate loss of data and control traffic, and adapt their operation to the network conditions. Their main difference is that SMT utilizes multiple paths simultaneously, in contrast to the single-path operation of SSP.

In this paper, we extend, refine, and analyze the operation of the two protocols. We present details and analyses of their mechanisms, including their interaction with the route discovery and the maintenance of multiple paths, the path-rating algorithm and a decision-theoretic model for the selection of its parameters, an algorithm to estimate the probability of path survival, and three alternative algorithms for automatic configuration of multipath transmissions. Last but not least, we evaluate the performance of SMT and SSP in a *realistic* network, integrating SMT and SSP with the *secure routing protocol* (SRP) [1], [17] and the IEEE 802.11 [18] as the data link protocol, and investigate the interaction of SMT and TCP.

Our experiments show that SMT and SSP can support applications with differing objectives and operate in a wide range of network conditions. The simultaneous usage of multiple paths and the dispersion of transmitted data enable SMT to support real-time traffic or other time-sensitive applications, even in highly adverse environments. In addition to highly reliable data delivery, SMT achieves low delay and low delay variability, at the expense of multipath transmission overhead. The overall overhead increase will be relatively low when real-time traffic comprises a small fraction of the overall network traffic. In resource-constrained environments, or when the supported application does not impose delay constraints, SSP is the appropriate lightweight alternative to SMT. SSP can provide secure and highly reliable data communication, trading off delay and delay variability for network overhead.

Moreover, we find that node mobility can be both detrimental and beneficial in adversarial environments, in contrast to the established belief that mobility impairs MANET communication. We also identify increased network load as a factor that can magnify the impact of attacks by relatively weakening the fault detection mechanisms. We combine SMT with TCP to provide flow control, and investigate their interaction: SMT thwarts malicious and benign faults, while TCP adjusts the end-to-end data rate according to the network conditions. Finally, we find that, with SMT, persistent disruption of the data transmission is more effective, from the adversary's point of view, than intermittent or "low-profile" attacks. Overall, our experiments show that SMT and SSP are versatile, effective, and efficient in a wide range of settings.

In the rest of this paper, we give a brief overview of the SMT and SSP, after introducing the network and security models. We present system components in Section IV, and the performance evaluation results in Section V, before a discussion of related literature and our conclusive remarks.

II. Network and Security Model

We define a network *node* as a process with: 1) a unique identity V; 2) a public/private key pair E_V, D_V; 3) a module implementing the networking protocols, e.g., routing, data transmission; and 4) a module providing communication across a wireless network interface. The combination of an *Internet protocol* (IP) address and a public key can uniquely identify a node.

We assume that any two nodes S and T that wish to communicate in a secure manner are capable to establish an end-to-end *security association* (SA). Since symmetric-key cryptographic primitives are computationally more efficient than public-key ones, we assume that a symmetric shared key $K_{S,T}$ instantiates the *SA* between the *end nodes*, the *source* S and the *destination* T. $K_{S,T}$ can be established through an authenticated Diffie–Hellman exchange [19] integrated with the initial route discovery [17]. Other methods to bootstrap associations are surveyed in [21]. We emphasize that the operation of SMT and SSP does not require that S and T are securely associated with any of the remaining, *intermediate* network nodes, which assist the S, T communication.

We make no assumptions on the behavior or the motivation of the intermediate nodes; they are either *correct*, that is, comply with the protocol rules, or *adversaries*, deviating from the protocol definition in an arbitrary manner. Adversaries can target the route discovery and the data transmission, corrupting, forging, or replaying routing, control, and data packets, mounting an attack either intermittently or persistently, in an attempt to control or deny communication.

We define a route as a sequence of nodes $\{V_0, \ldots, V_n\}$, which we denote as (S,T)-*route* when $S \equiv V_0$ and $T \equiv V_n$. The route discovery can be *explicit*, with the protocol returning the entire sequence of nodes, or *implicit*, with the protocol performing a distributed computation returning a (V_i, V_{i+1}, V_n)-tuple of the form (*current node*, *relay node*, *destination*) at each node $V_i \in (S,T)$-route, $i = 0, \ldots, n-1$. We assume that a secure routing protocol safeguards the route discovery, discarding erroneous connectivity information, and returning *correct routes*. A secure routing *specification*, that is, the sought properties for discovered routes, independently of the protocol operation, along with analyses of secure routing protocols, is given in [17] and [22].

III. Secure Data Transmission

A. Secure Message Transmission (SMT) Protocol

SMT uses an *active path set* (APS) comprising node-disjoint paths, determined and deemed operational at the source, for communication with a specific destination (Section IV-A). S *disperses* each outgoing message, adding limited redundancy to the data and dividing the resultant information into N pieces, which are transmitted across the APS routes one piece per route. Even if some of the message pieces are lost or corrupted, successful reception of M out of N pieces allows the reconstruction of the message at the destination. The ratio $r = N/M$ is termed the *redundancy factor*, and we denote a dispersed message with redundancy r as an (M,N)-message. Details and an example of the dispersion algorithm [23], which acts in essence as an erasure code, are given in [15].

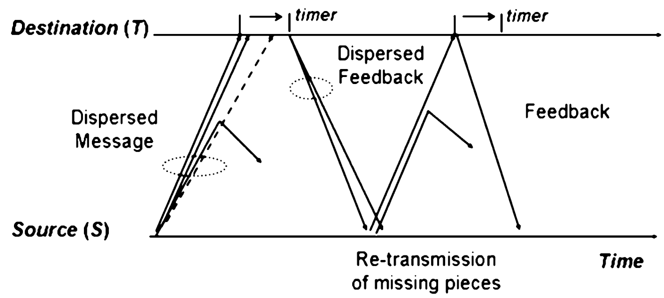


Fig. 1. SMT example: Transmission of a single message.

A *message authentication code* (MAC) [24], calculated with $K_{S,T}$, and a sequence number are appended to each piece, so that T can validate their integrity and origin authenticity, and reject replayed traffic. T reports successfully received pieces back to S, through cryptographically protected and dispersed feedback. S validates the feedback messages, unless a *retransmission timer* (RTO) expires when none of the message or feedback pieces are received.

While transmitting across the APS, S continuously updates the rating of the APS routes (Section IV-B). For each successful (failed) piece, the rating of the corresponding route is increased (decreased). A path is discarded once it is deemed failed, and a precaution is taken not to use the same path again if it is rediscovered within some time after it was discarded. While continuously assessing the quality of the utilized paths, and gathering statistical information on the network condition through the trusted destination feedback (Sections IV-B, IV-C), the protocol adjusts its configuration (Section IV-D), to remain effective in highly adverse environments and efficient in relatively benign conditions. In case the dispersed message cannot be reconstructed, T waits for S to retransmit the missing pieces, with a protocol-selectable maximum number of retransmissions, $\mathrm{Retry}_{\mathrm{MAX}}$, per serviced message.

An illustrative example of a message transmission is shown in Fig. 1. S disperses the message so that any three out of the four transmitted pieces are sufficient for successful reconstruction of the original message. Two of the pieces, each routed across a different route, arrive intact at the receiver, while the remaining two pieces are compromised by adversaries on the transmission paths; e.g., one piece is dropped and one (dashed arrow) is modified. The cryptographic integrity check reveals the corrupted data, T rejects the piece, and waits for additional message pieces (as determined in the header of incoming validated pieces), after setting a reception timer. At the timer expiration, the destination feedback is returned across the two operational paths. The sender receives and validates the feedback, ignoring duplicates, and retransmits the two missing pieces. One of them is lost, for example, because of intermittent malicious behavior, however, the destination has an adequate number of packets (3 out of 4), and acknowledges the successful reception to complete the message transmission.

B. Secure Single-Path (SSP) Protocol

SSP also provides data integrity, authenticity, and replay protection and it is equipped with the same end-to-end feedback and fault detection mechanisms as SMT [16]. Their main difference is that SSP transmits data across a *single* route, and does not perform data dispersion. In brief, SSP can be viewed as the limiting case of SMT, with a single-route APS. If provided with multiple, redundant routes, SSP utilizes them alternately. Accordingly, SSP provides feedback across a single route and switches to a new route only after the current one is deemed failed. Due to its single-path operation, SSP does not incur multipath transmission overhead and does not require the discovery of multiple routes, thus imposing in general less overhead than SMT. In the rest of this paper, we refer to SMT and SSP interchangeably, distinguishing between the two when necessary. We present components of the SMT and SSP protocols in Section IV, with Section IV-D and most of Section IV-A relevant to SMT only.

IV. System Components

A. Route Discovery and Determination of the APS

SMT and SSP are largely independent of the nature of the underlying secure routing protocol. Clearly, SMT requires that the protocol return multiple routes, while SSP can interoperate a protocol that returns a single route. They can both interoperate routing protocols performing either explicit or implicit route discovery (Section II). However, explicit route discovery, e.g., SRP [1], [7], [17] or SLSP [5], [17], or their combination, provides extensive control over the selection of the utilized routes. The source (sender) can compose routes in general different than those returned by the routing protocol, and, overall, implement a wide range of route selection algorithms. Moreover, it can unambiguously correlate loss/delivery of a packet with the route's constituent nodes (links) and, thus, avoid repeated attempts to communicate across the same yet nonoperational route. These two choices provide versatility and robustness.

An APS of k node-disjoint paths is constructed by successively calculating the node-disjoint, shortest in number of hops, paths [25], using the network connectivity information provided by the route discovery. The method's lack of sophistication is not a significant limiting factor, because the protocol does not have prior knowledge on the trustworthiness of individual network nodes. Intuitively, the selection of shortest routes is equivalent to the selection of the most secure paths, since any node can be initially considered equally probable to be an adversary.

Node disjointness enhances the robustness of SMT, because an adversary "strategically" situated on the overlap segment of two (or more) partially disjoint routes would control communication across those routes. Nevertheless, it is possible the protocol operates in conditions that allow only a few node-disjoint paths to be discovered, for example, due to a low-connectivity network topology or disruption of the route discovery by adversaries. In the extreme, SMT will operate as SSP.

The number of paths SMT should operate with depends on the protocol's configuration objective (Section IV-D), or it can be a protocol-selectable $\mathrm{APS}_{\mathrm{Threshold}}$ parameter. The richness of the APS, however, cannot be ensured or "required" from the underlying routing protocol. If available, a related route discovery parameter, such as SRP's number of route replies per query, can be tuned. In general, SMT attempts to determine new paths when new connectivity information is acquired, either proactively, or reactively, after the invocation of a route discovery.

However, route discoveries aiming to augment the APS must be rate-limited, to avoid repeated invocations when no additional (noncompromised) paths can be discovered. Finally, we note that placing the route selection at the sender implies that data are source-routed, functionality that is easy to combine with existing secure routing protocols.

B. Fault Detection

Each path in the APS is associated with a rating $r_s \in [r_s^{\text{thr}}, r_s^{\max}] \subset \Re$, with r_s^{thr} the minimum and $r_s^{\max}$ the maximum values of the path rating. The initial rating, assigned when a path is first added to the APS, is $r_s(0) = \delta(r_s^{\max} - r_s^{\text{thr}})$, with $0 < \delta < 1$. Equation (1) summarizes how the rating is updated after the ith transmission across a path that is not deemed failed yet, with constants $\alpha, \beta \in (0, r_s^{\max}]$

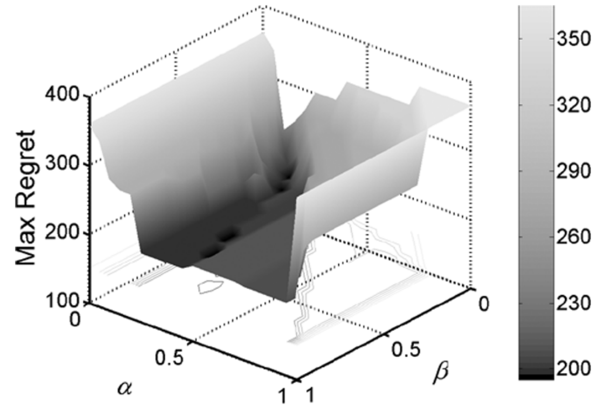
$$r_s(i) = \begin{cases} \max\left\{r_s(i-1) - \alpha, r_s^{\text{thr}}\right\}, & \text{if loss} \\ \min\left\{r_s(i-1) + \beta, r_s^{\max}\right\}, & \text{if success} \end{cases}. \quad (1)$$

This path rating scheme is not sensitive to the attack pattern an intelligent adversary selects. In general, an adversary could select an attack pattern in an attempt to maximize the number of packets it tampers with, while it remains undetected and the route it controls in use. We showed in [15] that for any arbitrary selection of packets the attacker tampers with, the *bandwidth loss* (BWL) over a path, that is, the fraction of discarded or corrupted packets without the route deemed nonoperational, is $\text{BWL} \le \beta/(\alpha + \beta)$.

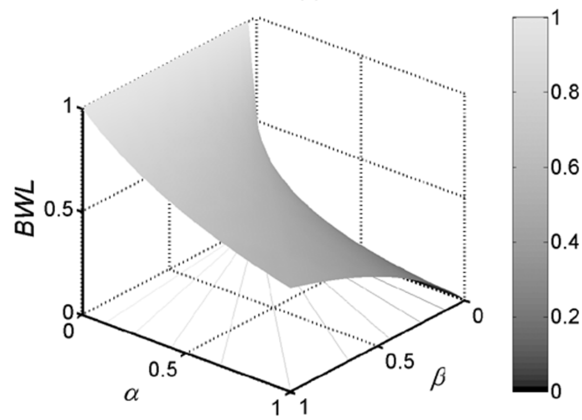
However, the performance of the protocol is sensitive to the α, β values. A criterion to determine the appropriate (α, β) values is necessary, depending on the *consequences* of the (α, β)-selection for any type of network and adversary. On the one hand, a relatively good path should not be discarded because of a short burst of errors. On the other hand, a poor path should be rejected as soon as possible, to minimize the packet loss and the resultant overhead. We model the selection of (α, β) values as a *decision theory* problem, with the most preferable pair of (α, β) values selected according to the *mini-max regret criterion* [26]. We present the decision-theoretic model and criterion in Appendix I, along with the proof of the Theorem, below, which summarizes the (α, β)-selection.

Theorem: If r_s is the rating of a path ρ, taking values in $[r_s^{\text{thr}}, r_s^{\max}] \subset \Re$, with $r_s^{\text{thr}} \ge 0$, and $\alpha, \beta \in (0, r_s^{\max}]$ two constants such that r_s is increased by β for each successful message reception and decreased by α for each failed message transmission across ρ, then α, β must be selected so that, for $k_1, k_2, k_3 > 0, \alpha - r_s^{\text{thr}} > k_1, r_s^{\max} - a > k_2$, and $\beta - r_s^{\text{thr}} > k_3$.

Fig. 2(a) shows the *maximum regret* for all (α, β)-selections as an example for a specific network setup. The formal definition is given in Appendix 1, but, intuitively, the *lower* the regret the *more preferable* is an (α, β) pair. Route reliability takes values $q_i \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, $r_s^{\text{thr}} = 0$, $r_s^{\max} = 1$, $r_s(0) = 0.75$, α, $\beta \in \{0, 0.05, 0.1, \ldots, 0.90, 0.95, 1\}$, the benefit from a delivered message is $P = 10$, and the costs of a message transmission and a route discovery are $C_1 = 1$ and $C_2 = 30$, respectively. A new route discovery is performed only when the utilized route is deemed failed, the data transmission stops after $L = 6$ discoveries if the Lth route is deemed failed, and $D = 40$ packets are transmitted. Fig. 2(a) confirms that very low or very high α values result in high regret (z axis), independently of the value of β. It also shows that low β results in high regret for a wide range of α values. In Fig. 2(b), BWL captures that with $\alpha \to 0$ the adversary can have full control over a path ($\text{BWL} \to 1$). BWL is also high when β is high, because the adversary can easily allow the path rating (and essentially its own credibility) to be reinstated. Finally, we observe in Fig. 2(a) a relatively wide range of (α, β) pairs that yield the same or almost the same low regret. This can be valuable, because (α, β) can be (re)selected randomly among those highly preferable pairs, making it hard for adversaries to infer the (α, β) values in use and adjust their attack pattern accordingly.



(a)



(b)

Fig. 2. Selection of the path rating protocol parameters. (a) Maximum regret and (b) BWL, shown as a function of α, β.

C. Path Survival Probability

We define the survival probability p_i of the ith path (route) in APS as the probability that the path will remain operational for the duration of the data transmission. We present here one algorithm to estimate p_i. We define the *lifetime* of the ith route, τ_i, as the time period from the discovery of the route until its removal from the APS, when the route is deemed nonoperational. If the route lifetime random variable is T_i, with cumulative distribution function $F_{T_i}(t) = \Pr\{T_i \le t\}$, then $p_i(t) = 1 - F_{T_i}(t) = \Pr\{T_i > t\}$ is the path survival probability, with t the current path *age*, measured in seconds.

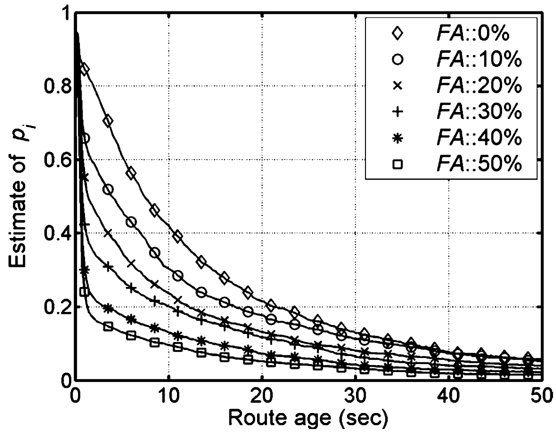


Fig. 3. Example of the average estimated path survival probability p_i, for indicative values of the route age t.

```
Input: k, [p1, p2,..., pk]
Output: R

R = zeros(k,k);
for m=1:k,
    for n=m:k,
        if ((m==1) and (n==1)),
            R(m,n)=pn;
        else if (m==1),
            R(m,n)=pn + (1-pn)R(m,n-1);
        else
            R(m,n)=pnR(m-1,n-1) + (1-pn)R(m,n-1);
        end
    end
end
```

Alg. 1. Calculation of R, the probability of successful delivery of an (M, N)-message across an APS of k paths, for all feasible M, N combinations.

The algorithm collects lifetime samples at the points in time it discards routes, with the s most recent samples, used to estimate the path survival probability at any time, $\tau_1, \tau_2, \ldots, \tau_s$, arranged in increasing order. If d is the transmission time of the longest possible piece, and t the age of the ith path, $p_i(t)$ is estimated by

$$\hat{p}_i(t) = \begin{cases} \frac{(s-1)}{s}, & \text{if } t+d < \tau_1 \\ \frac{(s-j)}{s}, & \text{for } j \text{ s.t. } \tau_j \leq t+d < \tau_{j+1} \\ \frac{1}{s}, & \text{if } t+d > \tau_s \end{cases} \tag{2}$$

Equation (2) extends the estimator in [27] when $t + d \notin [\tau_1, \tau_s]$, to avoid overestimating or underestimating the path survival probability. If $t + d < \tau_1$, (2) yields $\hat{p}_i(t) \neq 1$, thus, the protocol avoids utilizing only one, newly calculated yet possibly compromised route. If $t + d > \tau_D$, (2) yields $\hat{p}_i(t) \neq 0$, thus, the protocol does not ignore a long-lived but valid route.

The protocol should be allowed time to collect the initial window of s samples before it computes its first estimate; during this time, all available routes are assigned some arbitrary (and equal for all routes) survival probability. In Fig. 3, we show the average estimate of p_i, calculated from data collected from the experiments in Section V, as a function of the route age and for different values of the *fraction of adversaries* (FAs) present in the network. As FA increases, the discovery of compromised routes becomes more frequent, resulting in decreased probability estimates. Moreover, the older the route is, the lower is its probability of survival.

D. Multipath Configuration

The protocol automatically adapts the configuration of the message transmission, determining the redundancy factor and which and how many of the APS routes to utilize. The inputs to the configuration algorithm are: 1) an APS of k paths; 2) their ratings r_s^j; and 3) their survival probabilities, p_j, for $1 \leq j \leq k$; 4) an optimization objective; and 5) an objective-specific parameter, i.e., P_{GOAL}, the sought probability of successful message delivery, or, r_{GOAL}, the maximum allowable redundancy. The algorithm outputs are: 1) N and M, with $1 \leq M \leq N \leq k$ and 2) the indices $i_1, i_2, \ldots, i_N$ of the N APS paths to be utilized for the (M, N)-message transmission.

We denote the probability of successful reception of an (M, N)-message as $R(M, N)$. We consider the following objectives:

$$P_{\mathrm{GOAL}} - N_{\min} : \min_{\substack{1 \leq N \leq k \\ 1 \leq M \leq N}} \{N | R(M, N) \geq P_{\mathrm{GOAL}}\} \tag{3}$$

$$P_{\mathrm{GOAL}} - r_{\min} : \min_{\substack{1 \leq N \leq k \\ 1 \leq M \leq N}} \{r | R(M, N) \geq P_{\mathrm{GOAL}}\} \tag{4}$$

$$r_{\mathrm{GOAL}} : \max_{\substack{1 \leq N \leq k \\ 1 \leq M \leq N}} \{R(M, N) | r \leq r_{\mathrm{GOAL}}\}. \tag{5}$$

$P_{\mathrm{GOAL}} - N_{\min}$ seeks the lowest number of paths N such that $R(M, N)$ is at least equal to P_{GOAL}; it ensures resistance to failures, while keeping the multipath overhead and in particular the number of message pieces low. $P_{\mathrm{GOAL}} - r_{\min}$ seeks to achieve P_{GOAL} with the lowest redundancy but without constraining N. Finally, r_{GOAL} seeks to maximize under the constraint that r does not exceed r_{GOAL}, to limit the transmission overhead.

The configuration algorithm ranks the paths in decreasing order of their ratings, r_s^j, then, for equally rated paths, in decreasing order of their survival probabilities p_j, and, finally, as a tiebreaker, in increasing order of their hop length. Then, Alg. 1 calculates efficiently[1] all the $R(M, N)$ values for this ordering of paths and their corresponding $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ values. A search in $R(M, N)$ is performed to determine the pair of M, N values that satisfy the objective. For (3), a search along the first row yields N_{min} such that $R(1, N_{\min}) \geq P_{\mathrm{GOAL}}$, and a search along the $N_{\min}$th column checks if $M > 1$ exists so that $R(M, N_{\min}) \geq P_{\mathrm{GOAL}}$. For (4), while maintaining the lowest r that yields $R(M, N) \geq P_{\mathrm{GOAL}}$, it suffices to search each column until $R < P_{\mathrm{GOAL}}$, because, for a fixed N, $R(M, N)$ is a decreasing function of M, and then move to the next column. If $R(M, N)$ does not satisfy the objective for any M, N, then the values and corresponding paths that simply maximize $R(M, N)$ are returned. For (5), the column-wise search continues till $N/M > r_{\mathrm{GOAL}}$, while keeping track

[1]With each message piece sent across a distinct path, a transmission is successful if any set of M out of N paths, is operational. In other words, $R(M, N)$ is the probability that *at least one* of those minimal path sets is operational. Due to the path disjointness, the calculation of $R(M, N)$ is straightforward, using the inclusion–exclusion principle [29] or an improved method [30]. A more efficient solution is Alg. 1, based on [31] and [32]: $k^2/2$ $R(i, j)$ values are calculated for all $i \leq M, j \leq N$, with $O(k^2)$ time and memory complexity.

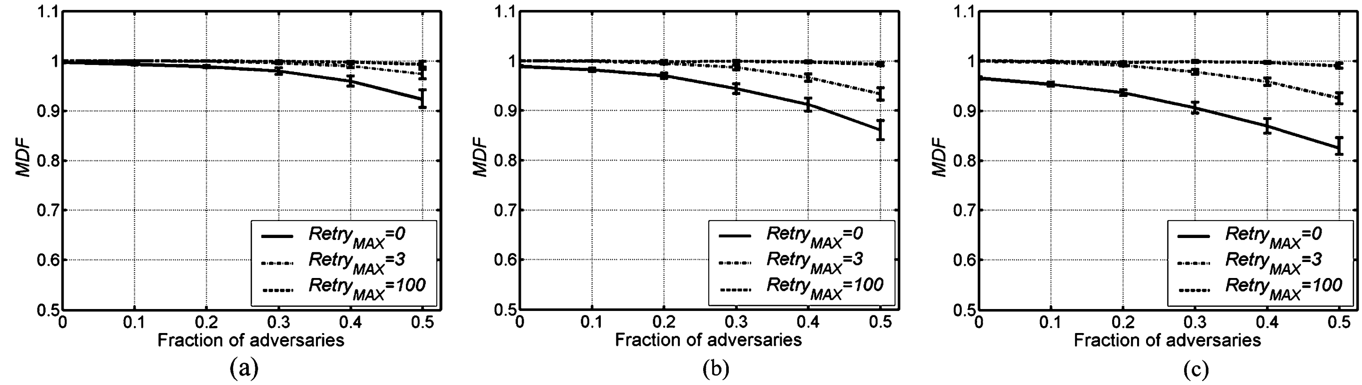


Fig. 4. Message delivery fraction (MDF). (a) SMT-LS. (b) SMT-RRD. (c) SSP.

of the maximum R. The (M,N)-message configuration is determined independently for each message, as p_j, r_s^j, or the APS itself, change over time.

If the constraint in (3) and (4) cannot be satisfied, then additional, new paths are required to enhance the likelihood of successful reception. A more fine-grained decision on whether new paths are necessary can be based on the condition of the currently utilized N paths. The criterion, inspired by the *reliability importance* of a component in a system [28], is the importance of the not-yet-discovered $(N+1)$st path, with p_{N+1} denoting its survival probability

$$I_{N+1} = \frac{\partial R(M, N+1)}{\partial p_{N+1}} = R(M-1, N) - R(M, N). \quad (6)$$

I_{N+1} is the probability that exactly $M-1$ of the N paths are operational, or in other words, the probability that the addition of the $(N+1)$st path will allow the $(M, N+1)$ transmission to be successful. I_{N+1} can be easily calculated from the R matrix.

V. Performance Evaluation

We implemented OPNET simulation models of SMT and SSP, with a 1000 m $\times$ 1000 m network coverage area and 50 nodes. The 802.11 [18] is the data link layer protocol. The nominal communication range of 300 m yields biconnected network topologies with high probability [33]. The data rate is 5.5 Mb/s, the buffer size at the medium access control layer is 655 Kbits and three constant-bit-rate (CBR) sources generate four messages/s, with message size of 512 bytes. The buffer at the network layer was not a limiting factor; i.e., no packets were lost due to overflow at the source node. Each source is securely associated with one destination, transmitting data to the same destination throughout the simulated period of 900 s. The nodes are initially uniformly distributed throughout the network area and their movement is determined by the *random waypoint mobility model* [34], with nodal speed uniformly distributed between 1 m/s and 20 m/s [35]. Each point on the presented graphs is the average of 30 randomly seeded runs with 90% confidence intervals shown. The number of adversaries is 0, 5, 10, 15, 20, and 25, and results are presented as a function of the *fraction of adversaries* (FA) in the network, i.e., 0%, 10%, 20%, 30%, 40%, and 50%, respectively. Adversaries disrupt the data transmission, tampering with in-transit packets, persistently in Sections V-A–V-E, and intermittently in Section V-F. The protocol parameters are set to $P_{\text{GOAL}} = 0.95$, $r_s^{\text{thr}} = 0.0$, $r_s^{\max} = 1.0$, $\alpha = 0.33$, $\beta = 0.033$, $\delta = 0.75$, unless noted, otherwise.

We first examine SMT when routes are provided by an idealized discovery scheme with no delay and no control overhead. This *SMT with link-state* (SMT-LS) instantiation allows us to isolate the performance of SMT from the underlying routing protocol, since different routing protocols may impose different limitations (e.g., number of available routes), different delays, and different amounts of overhead. Then, as a practical scenario, we examine SMT and SSP in conjunction with the *secure routing protocol* (SRP) [1], [17], denoting this SMT instantiation as *secure message transmission with reactive route discovery* (SMT-RRD).

The evaluated performance metrics are: 1) the message delivery fraction (MDF), i.e., the ratio of the number of received messages to the number of transmitted messages; 2) the average message delay (D), i.e., the period from the arrival (generation) of a message at the source, until its reception at the destination; 3) the delay jitter (J), i.e., the delay variability around the average delay; 4) the transmission overhead $(TXOV)$, i.e., the fraction of all bits transmitted by SMT and SSP, including redundancy, protocol headers, and feedback, over the total of successfully received data bits; and 5) (where applicable) the routing overhead ROTV, i.e., the fraction of all the bits transmitted (counted at each network hop) by the underlying routing protocol, over the total of data bits successfully received at their destination.

A. Reliability

SMT-LS, SMT-RRD, and SSP achieve highly reliable communication, even when FA $= 50\%$. All three protocols can promptly detect nonoperational paths, avoid them, and communicate across those supporting reliable communication. SMT can minimize the number of retransmissions, since dispersion across multiple paths masks data loss. The larger the number of available paths, the larger the fault-tolerance, this being especially true for SMT-LS that operates with an APS of large cardinality (an average of 8.5 paths). Fig. 4(a) shows that without retransmissions $(\text{Retry}_{\text{MAX}} = 0)$ SMT-LS delivers more than 99% of the data for FA up to 20%, and experiences loss of 2% to 7% of the messages for FA from 30% to 50%. On the other hand, the partial view of the network connectivity restricts SMT-RRD,

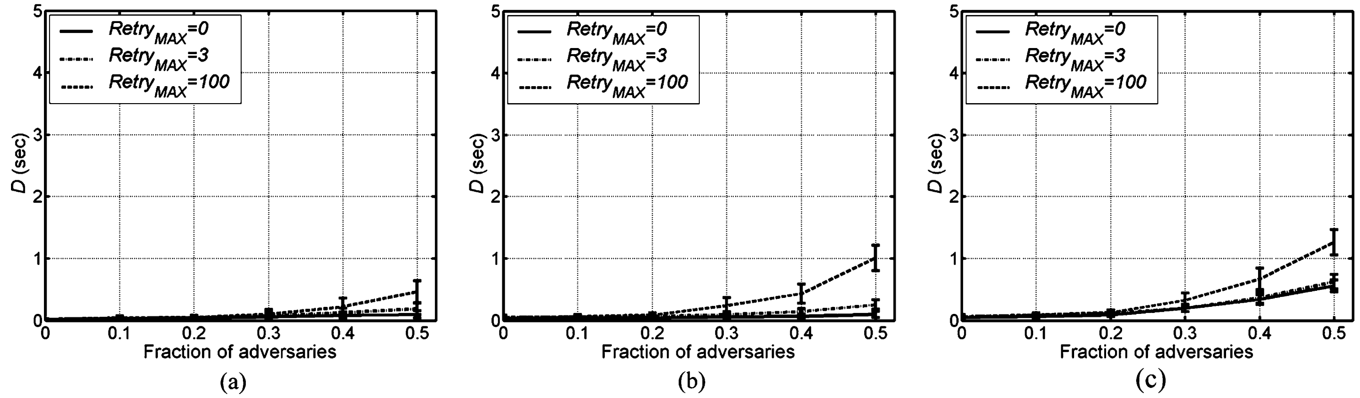


Fig. 5. End-to-end message delay. (a) SMT. (b) SMT-RRD. (c) SSP.

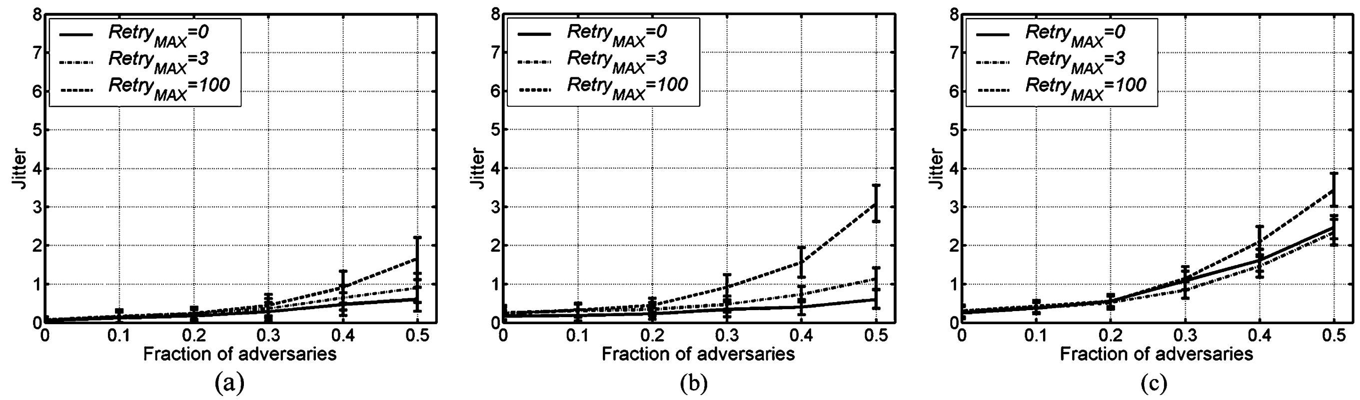


Fig. 6. End-to-end message delay variability (jitter). (a) SMT-LS. (b) SMT-RRD. (c) SSP.

which operates with an APS of 3 node-disjoint paths with a standard deviation of 1.5 paths, while SSP discovers and utilizes only a single route. Yet, SMT-RRD remains competitive with SMT-LS, with 1% to 7% lower MDF [Fig. 4(b)], while MDF for SSP is 3.1% to 10.3% lower for no retransmissions.

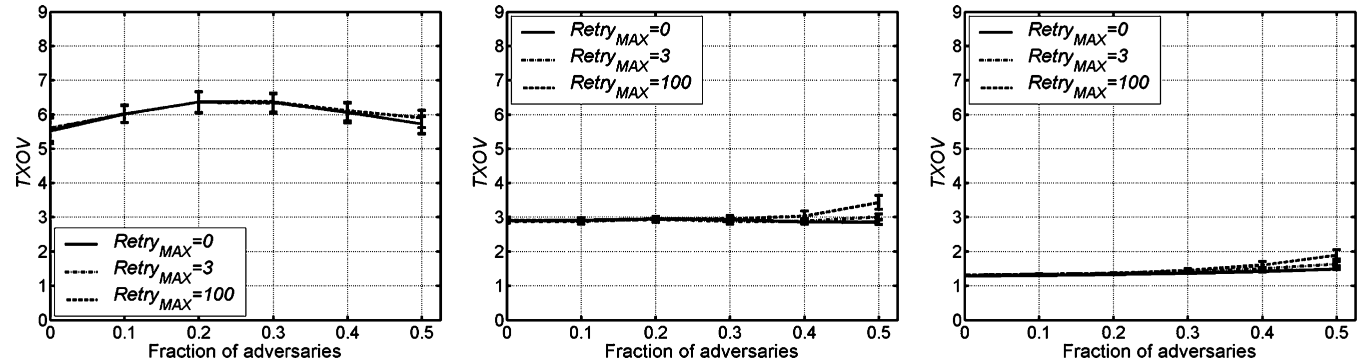
Retransmissions can mitigate this degradation to achieve highly reliable communication. For a maximum of three retransmissions per message, $(\mathrm{Retry}_{\mathrm{MAX}} = 3)$, SMT-LS delivers more than 97% of the messages even when $\mathrm{FA} = 50\%$. The improvement is more significant for SMT-RRD, which delivers only up to 2.3% to 4% fewer messages than SMT-LS $(\mathrm{FA} = 40\%, 50\%)$. SSP is only 1% less reliable than SMT-RRD on the average. The differences between the protocols become imperceptible when retransmissions fully correct data loss $(\mathrm{Retry}_{\mathrm{MAX}} = 100)$. (Note: Messages with delays above 30 s are ignored; up to 0.7% of the messages are not accounted for in Fig. 5)

B. Delay and Delay Variability

Fig. 5(a) and (b) shows that SMT is capable of achieving low delay, which remains nearly constant for networks with up to 20% adversaries and increases only slowly as FA increases further. This is expected when no retransmissions are used and paths are readily available SMT-LS, but we find that this is also the case for SMT-RRD and $\mathrm{Retry}_{\mathrm{MAX}} = 3$. Accordingly, the delay variability shown in Fig. 6(a) and (b), increases slowly for both SMT-LS and SMT-RRD, with the former achieving 11%–75% lower jitter and 1%–64% lower delay than the latter, for $\mathrm{Retry}_{\mathrm{MAX}} = 0$ or 3. Delay and jitter are low due to multipath transmission and the resultant scarce use of retransmissions; e.g., for $\mathrm{FA} = 20\%$, SMT-RRD retransmits (fully or partially) only 6.8% and SMT-LS only 3.4% of the total transmissions.

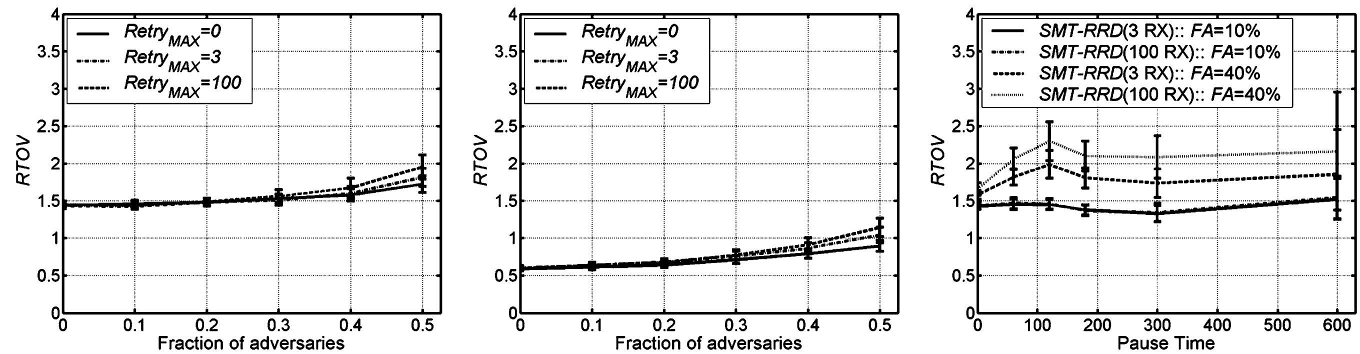
This clues the ability of SMT to support real-time traffic even in the presence of a significant fraction of adversaries, greatly facilitated by its low-cost and effective path "testing": multipath transmissions allow for the rating of multiple paths to be updated simultaneously with the actual data forwarding, while use of low-rated paths does not result in message loss due to the data dispersion.[2] It is important that even with all but one of the N paths compromised, a single feedback message allows SMT to update the rating of those paths without a time-out. In contrast, SSP relies solely on timeouts to detect transmission failures, and it is inherently slower in detecting a compromised route. For example, with no retransmissions, the delay for SSP [Fig. 5(c)] is 38%–480% larger than the delay for SMT-RRD and 99%–486% higher than the delay for SMT-LS. Similarly, the jitter for SSP [Fig. 6(c)] is 50%–307% higher than SMT-RRD and 208%–380% higher than SMT-LS. As the number of retransmissions increases, SSP experiences up to 155% and 237% higher delays, and up to 105% and 237% higher jitter than SMT-RRD and SMT-LS, respectively.

[2]We emphasize that the protocols do not assume knowledge on the trustworthiness of individual nodes or their misbehavior patterns, and they do not rely on "testing." Rather, they transmit actual data, while determining the condition of the APS paths.



(a) (b) (c)

Fig. 7. Transmission overhead (TXOV). (a) SMT-LS. (b) SMT-RRD. (c) SSP.



(a) (b) (c)

Fig. 8. Routing overhead (RTOV). (a) SMT-RRD as a function of FA. (b) SSP as a function of FA. (c) SMT-RRD as a function of PT.

The case of $\mathrm{Retry}_{\mathrm{MAX}} = 100$ has a special interest, as the delay and the jitter [Figs. 5 and 6] are then significantly larger for all the protocols, compared with $\mathrm{Retry}_{\mathrm{MAX}} = 0$ or 3. We observed in Fig. 4 that these additional retransmissions offer up to 7% higher MDF for SSP, up to 6.3% for SMT-RRD, and up to 2% for SMT-LS, when $\mathrm{FA} = 50\%$. However, this comes at the comparatively large cost of increasing the delay from 102% to 290%. The reason becomes apparent by examining, in comparison, traces of communication for one of the flows for $\mathrm{FA} = 10\%$ and 50%; in the latter case, nodes may get disconnected for long periods of time, with messages backlogged during such periods of disconnection and delivered only after repeated (and possibly many) retransmissions.

C. Overhead and Multipath Configuration

The discussion above demonstrated that the availability of many paths can be beneficial. However, this comes at the cost of increased network overhead. Fig. 7 shows that SMT-LS has the largest and SSP the smallest overhead, with the overhead being roughly proportional to the APS cardinality. (Note: The SMT header is 22 bytes long and the SMT feedback size is 19–35 bytes, both including an 80-bit HMAC [24].) SSP imposes transmission overhead which is 54% smaller than that of SMT-RRD and 78% smaller than that of SMT-LS [Fig. 7] and 73% smaller routing overhead than SMT-RRD [Fig. 8(a) and (b)]. This implies that, essentially, SSP trades off delay and delay variability for transmission and routing overhead.

Next, we examine the three configuration algorithms from Section IV-D with SMT-LS, to allow the algorithms to operate over a wider range of parameters; results for N and M are shown in Fig. 10(a) and (b), respectively. $P_{\mathrm{GOAL}} - N_{\min}$ utilizes on the average five paths while requiring the successful delivery, practically at all times, of one message piece. For high P_{GOAL} and p_i not close to 1, it highly unlikely to have $M > 1$ such that $R(M, N_{\min}) > P_{\mathrm{GOAL}}$. In comparison, $P_{\mathrm{GOAL}} - r_{\min}$ utilizes up to 35% more paths, while the average transmission redundancy is up to 9% smaller, because a larger M is chosen. $P_{\mathrm{GOAL}} - r_{\min}$ does not achieve an impressive reduction in the redundancy, because P_{GOAL} is set to a large value. But, in general, it trades off fault-tolerance (higher M), for more effective path "testing" (higher N) at the same or smaller cost. The combination of this testing with the use of highly rated paths, even when their p_i is small, allows $P_{\mathrm{GOAL}} - r_{\min}$ to achieve 18% to 33% lower delay than $P_{\mathrm{GOAL}} - N_{\min}$, while achieving the same MDF.

For both $P_{\mathrm{GOAL}} - r_{\min}$ and $P_{\mathrm{GOAL}} - N_{\min}$, N decreases as FA increases, even though one would expect the opposite. However, a large fraction of adversaries implies it is more likely that any discovered route will be compromised. SMT may start with a large APS, but it soon discards the compromised routes, and operates with the remaining operational ones. We confirmed this from the histogram of the utilized APS sizes, which is shifted and becomes more concentrated toward lower values as FA increases.

In Fig. 10, we also observe that the r_{GOAL} configuration, for which we chose $r_{\mathrm{GOAL}} = 2.5$, utilizes on the average $N = 3$ paths, and r slightly above 2. r_{GOAL} achieves from 6% to 25% smaller delay than $P_{\mathrm{GOAL}} - r_{\min}$ and 37% to 42% smaller than

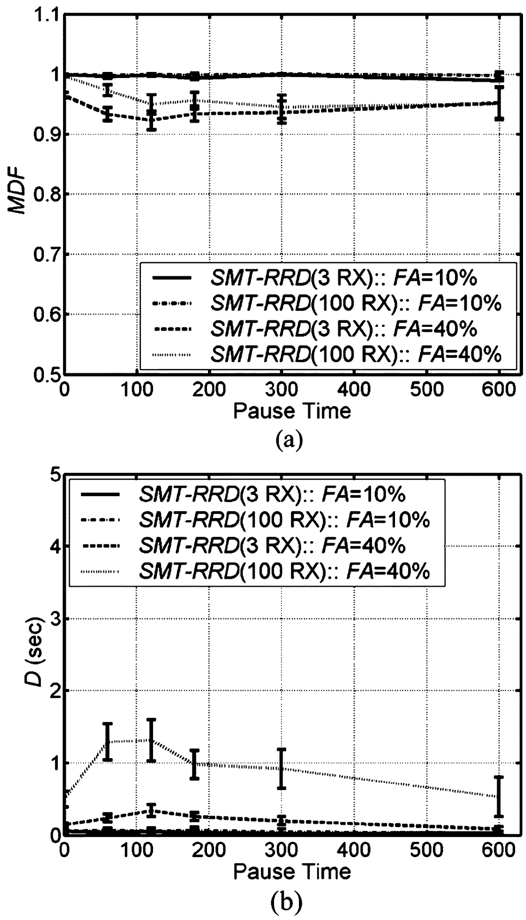


Fig. 9. Impact of mobility; SMT-RRD. (a) Message delivery fraction (MDF). (b) End-to-end message delay (D).

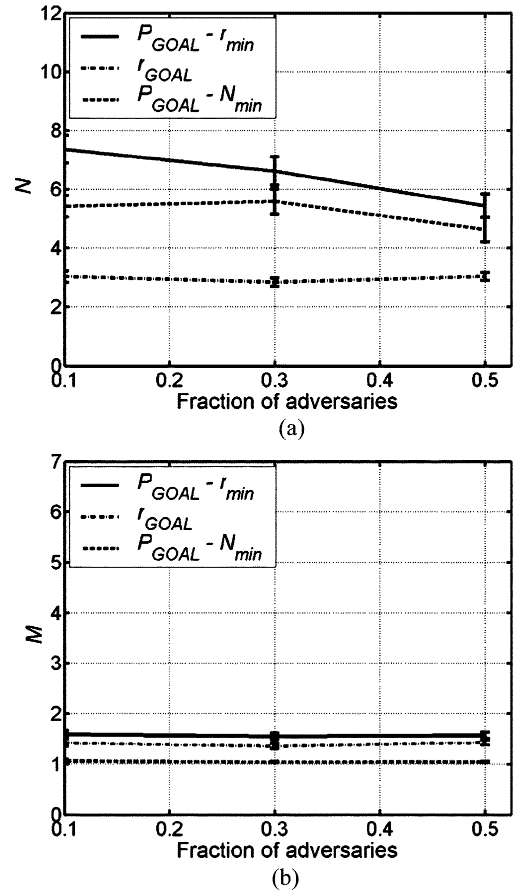


Fig. 10. Transmission redundancy. (a) Average number of sent pieces (N). (b) Average number of required pieces (M).

the $P_{\mathrm{GOAL}} - N_{\min}$. This clues that even a relatively limited number of paths and redundancy can yield effective communication, very similarly to SMT-RRD. Finally, Fig. 11(a) shows the average R decreasing with FA increasing, as expected, and Fig. 11(b) shows, for the $P_{\mathrm{GOAL}} - N_{\min}$ algorithm, R "following" the protocol-selectable parameter P_{GOAL}.

D. Mobility

Mobility is modeled in our simulation by the *pause time* (PT); for low mobility, PT is high and *vice versa*. We find that when the fraction of adversarial nodes is relatively small ($\mathrm{FA} = 10\%$), data loss due to route breakages and due to adversaries is masked by SMT-RRD, and mobility has a small impact on the protocol performance. This is demonstrated in Fig. 9, with the protocol achieving almost perfectly reliable communication without an increase in delay. In contrast, mobility has higher impact for a large fraction of adversaries ($\mathrm{FA} = 40\%$). As mobility decreases, MDF decreases up to 5% and the delay increases up to 153% ($\mathrm{PT} = 120$ s), and then MDF and delay gradually increase and decrease, respectively, as PT increases further. Eventually, for $\mathrm{PT} = 600$ s, MDF is 1% to 4% lower than MDF for $\mathrm{PT} = 0$ s, while the delay and jitter are up to 42% and 21% smaller, respectively.

These observations indicate that mobility can be both beneficial and detrimental to the protocol operation. When no operational paths are available, reconfiguration of the network due to mobility can restore connectivity. On the other hand, mobility causes breakage of routes that are free of adversaries, possibly adversely impacting the connectivity. As a result, both the delay and the reliability of the communication may deteriorate. The first trend explains why MDF is larger for higher mobility and is reduced as mobility decreases ($\mathrm{PT} = 120$ s), and the second trend explains why delay decreases as mobility decreases further. However, MDF remains slightly smaller for 600 s than for 0 s, because a small percentage of transmissions suffer long disconnection periods, while most of the transmitted data undergo very few interruptions. The low mobility ($\mathrm{PT} = 600$ s) emphasizes both tendencies, causing larger spread of the delay, and especially the routing overhead [Fig. 8(c)], with pronounced RTOV variability due to the repeated route discoveries by disconnected nodes.

E. Load and Interaction With TCP

In Sections V-A–V-D, we evaluated SMT with constant bitrate data flows. We examine now its performance as the offered load increases. Note that SMT does not provide flow control mechanisms, which is a functionality of transport layer protocols, such as TCP. We increase first the load offered by the CBR sources, up to the point the average end-to-end throughput cannot increase further, for $\mathrm{FA} = 10\%$ and constant mobility ($\mathrm{PT} = 0$ s). In Fig. 12(a), the network is saturated when 16 messages/s per flow are transmitted ($\mathrm{MDF} = 90\%$), with the degradation already visible for 14 messages/s ($\mathrm{MDF} = 96\%$).

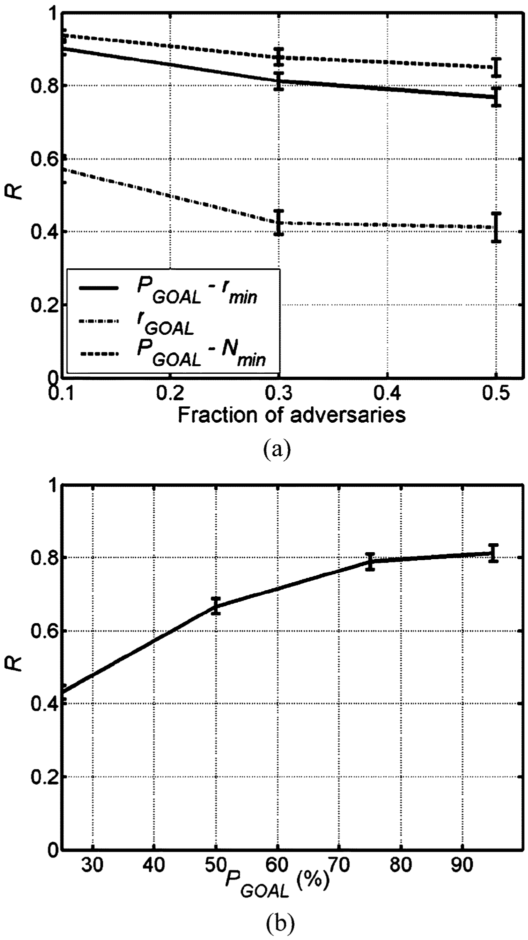


Fig. 11. Average $R(M_{\rm opt}, N_{\rm opt})$ as a function of: (a) fraction of adversaries. (b) $P_{\rm GOAL}$, for $P_{\rm GOAL} - N_{\rm min}$ objective.

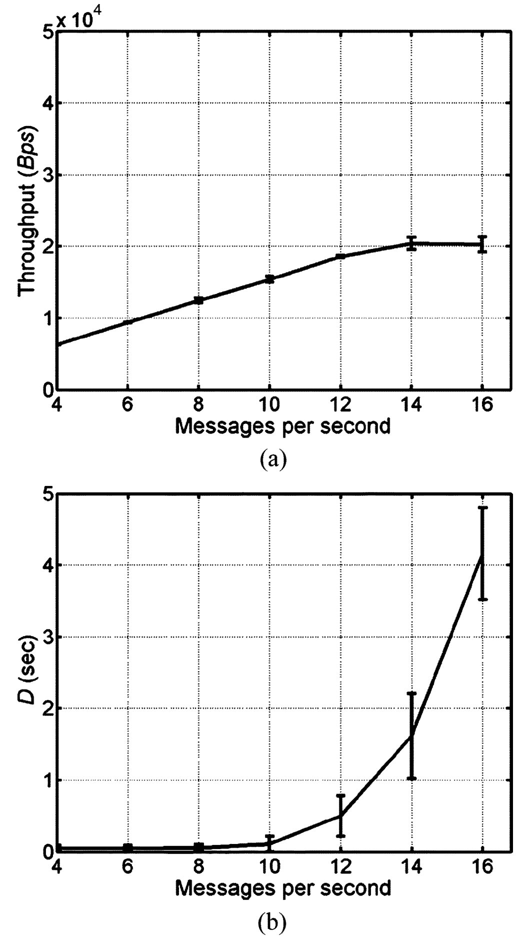


Fig. 12. SMT-RRD performance as a function of network load. (a) Average throughput. (b) End-to-end message delay (D).

Fig. 12(b) is more suggestive, with the delay increasing fast for rates above 12 messages/s. A close look at the data link layer confirms that the network becomes gradually more congested; the delay to access the medium increases by 218% and the route errors (i.e., lost hopwise transmissions) increase by 260%. SMT can tolerate such failures, but only partially as the network gets saturated. Not only does the network impair communication, but the impact of adversaries also becomes more severe. In a heavily loaded network, the RTO estimate will be set inevitably to a high value to reflect the increased round-trip times. High RTO implies slow detection of transmission failures, which, in turn, implies higher message loss across compromised paths. In our experiment, as compared with the transmission of 10 messages/s, the fraction of message pieces that adversaries drop is tripled when 16 messages/s are transmitted, with the adversary disrupting 4% and 14% of the SMT message transmissions, respectively.

Next, we integrate SMT with TCP, and have the communication data rate regulated by the TCP flow control mechanism. We establish ftp sessions with the source "uploading" 1.5 MB files to the destination, one after another. We use the default settings for TCP, the New Reno flow control mechanism, and time-based TCP retransmission thresholds. We select to utilize SMT-RRD with $\mathrm{Retry}_{\mathrm{MAX}} = 3$, so that SMT and TCP have mostly distinct roles: on the one hand, SMT is responsible for adapting and configuring transmissions to mitigate malicious and benign failures, possibly with limited fast retransmissions. On the other hand, TCP is responsible for retransmitting all data and especially those that SMT did not deliver. It is expected that TCP and SMT retransmissions do not overlap ($\mathrm{RTO}_{\mathrm{min}}^{\mathrm{TCP}} = 0.5$ s and $\mathrm{RTO}_{\mathrm{min}}^{\mathrm{SMT}} = 0.1$ s), while SMT frequently uses partial retransmissions, without RTO expiring.[3]

We measured the performance of the combined TCP-SMT-SRP system in the same mobility scenarios of the increased CBR load, as well as in a benign (FA $=$ 0%) and in a more adverse (FA $=$ 30%) environment. As a point of reference, we also evaluated TCP combined with SRP and no data transmission security for the same set of scenarios. Fig. 13(a) shows that with the data rate controlled by the TCP flow control, the average end-to-end throughput achieved is approximately equal to 12 messages/second per flow, while the end-to-end delay [Fig. 13(b)] is 56% lower than the delay for CBR flows with the same rate. TCP "estimates" the available bandwidth, which in the case of malicious environments reflects the operation of the underlying SMT that tolerates faults. While SMT successfully masks data loss and prevents retransmissions at the transport layer, TCP increases the rate of transmitted

[3]Depending on the algorithm, the fractions of partial SMT retransmissions are: 11%–18% for $P_{\rm GOAL} - N_{\rm min}$, 37%–56% for $P_{\rm GOAL} - r_{\rm min}$, and 14%–47% for $r_{\rm GOAL}$.

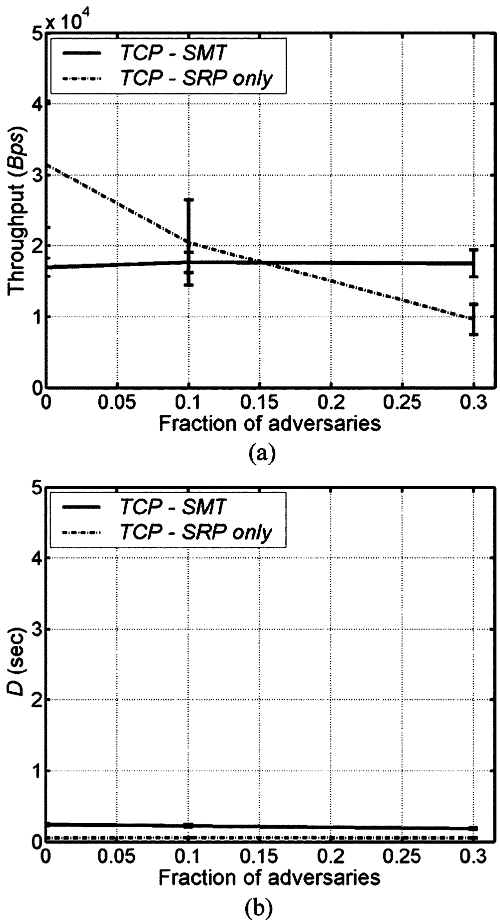


Fig. 13. Integration of SMT-RRD with TCP. (a) Average throughput. (b) End-to-end message delay (D).

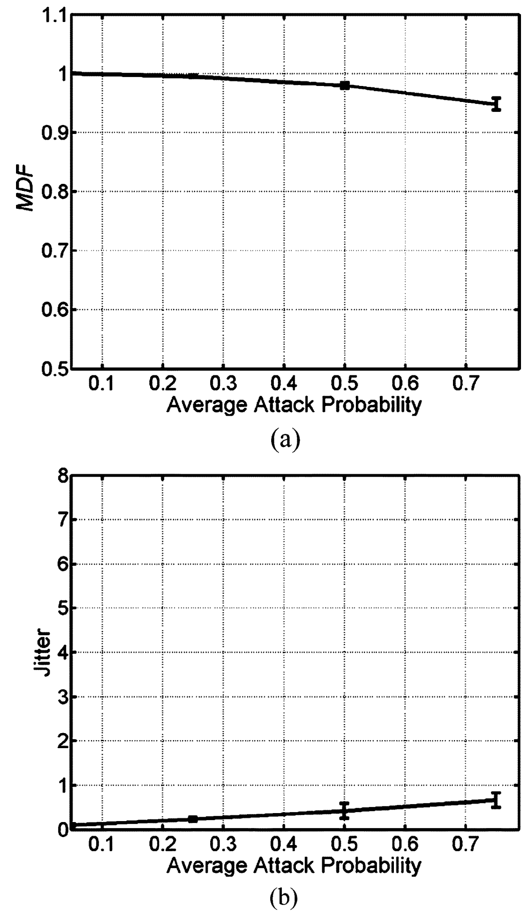


Fig. 14. SMT-RRD performance as a function of the attack pattern. (a) MDF. (b) Delay variability (jitter).

data. When adversaries, route breakages, or congestion causes data loss that SMT cannot tolerate, TCP inevitably enters congestion avoidance or returns to the slow-start phase, while SMT recovers, for example, by reconfiguring transmissions across alternative routes.

The ability of SMT to safeguard the flow of data is apparent as the FA increases from 10% to 30% and the throughput remains practically constant. In contrast, when no security for the data transmission is available, the end-to-end throughput drops abruptly as the fraction of adversaries increase. Along a compromised path, an adversary can drop a fraction of in-transit data or feedback messages and still force the flow of data at a very low rate; only if a route error occurs, data will be rerouted across an alternative route.[4]

With $\mathrm{FA} = 10\%$, we observe that TCP-SRP achieves 15% higher throughput than TCP-SMT on the average but with 250% higher variability. When the path is compromised, the TCP-SRP throughput is practically zero, but, otherwise, it can be twice as large as that of TCP-SMT when no malicious disruptions occur. SMT's roughly threefold transmission overhead, as compared with the single-path SRP forwarding, is the reason for TCP-SMT achieving 46% smaller throughput when $\mathrm{FA} = 0\%$.

[4]We did not implement it here, but an adversary can always hide a route breakage at a downstream link by discarding route error packets.

F. Variable Attack Patterns

In all experiments above, the adversaries dropped all in-transit data packets. We do not present results with adversaries corrupting data or SMT feedback, which are detected at the destination and source, respectively, since these attacks did not produce any significant difference. We conjecture that persistent disruption of the data transmission is the worst form of attack, as it has the most severe impact, compared with attackers that intermittently tamper with data. Clearly, an intelligent intermittent attacker could selectively discard data and still degrade communication. However, the dispersion of data across multiple paths makes it hard for adversaries to be effective: it would require that at least M of the N pieces were dropped by M coordinated adversaries. It is, nevertheless, interesting to investigate the impact of intermittent misbehavior, which could allow an adversary to remain on a utilized route.

We evaluate SMT-RRD with $\mathrm{FA} = 50\%$ and each adversary dropping data with a probability q_{DROP}, which has average value $q_{\mathrm{AV}} \in \{0.05, 0.25, 0.5, 0.75\}$ and it is selected uniformly from the (0, 0.1), (0, 0.5), (0.25, 0.75), (0.5, 1) intervals, respectively. In all runs, SMT uses the same parameters $(\alpha, \beta) = (0.3, 0.05)$ and $\mathrm{Retry}_{\mathrm{MAX}} = 3$. Fig. 14(a) shows that MDF decreases slowly; for example, as q_{AV} increases from 0.25 to 0.5, the fraction of data packets dropped by the adversary increases by 41% but MDF decreases only by 1.5%.

Nevertheless, it is obvious that performance, shown here by MDF and jitter [Fig. 14(b)], deteriorates as attackers become more persistent.

VI. Related Work

The use of multiple paths has been widely studied for the provision of QoS guarantees and load balancing in wired networks. In MANET, multiple paths have been utilized as a means to tolerate path breakages due to mobility. Reference [12] proposes the collection of link quality (reliability) metrics and the fast determination of a set of highly reliable, and thus, long-lived, link-disjoint paths (as opposed to node-disjoint paths that we use here). References [13] and [14] propose the use of diversity coding and provide an approximation for the probability of successful data transmission. None of the two above-mentioned schemes provides security features or mechanisms to assess the quality of utilized routes in an end-to-end manner.

A number of works secured the MANET route discovery. Beyond SRP, [2] proposed a secured version of the ad hoc on-demand distance vector (AODV) routing protocol [36], using public key mechanisms to authenticate the end to intermediate nodes and set forward and backward routes, and a hash-chain to prevent adversaries from decreasing the route hop count. Reference [3] proposed a protocol to secure the dynamic source routing (DSR) protocol [37], using symmetric-key primitives and time synchronization to authenticate the nodes of the discovered routes. Reference [4] uses public-key primitives for a secure routing protocol that resembles AODV, with simplified functionality. Reference [6] is a secure proactive distance-vector routing protocol, [5] is a secure link-state protocol discovering network connectivity within a zone [38] of x hops, and [39] proposes security mechanisms for the optimized link-state routing protocol (OLSR) protocol [40].

As for MANET data transmission security, the use of multiple routes was first proposed in [20] and [1]. Reference [41] proposed an "onion encryption"-based mechanism to detect a faulty link, with one of its incident nodes disrupting the data transmission, if data loss along a route drops below a tolerable threshold. From a different perspective, [42] proposed to detect misbehaving nodes by local monitoring and disseminate alerts for such events, so that routes through relatively well-behaved peers are selected. Reference [43] seeks to isolate misbehaving nodes, and also relies on dissemination of misbehavior reports, with the provision of authenticating those messages. Reference [44] proposes to stimulate cooperation of rational nodes through fictitious currency and remuneration, and a game-theoretic motivation for a reputation system is proposed in [45].

Unlike previous works, SMT provides a solution tailored to MANET environment, combining four elements: 1) reliance only on end-to-end security bindings; 2) simultaneous transmission across multiple, diverse routes determined by the protocol; 3) robust detection of communication faults; and 4) adaptation to the network condition. Such or similar features were proposed individually, e.g., [10], [12]–[14], and [46], but they were not combined before into one protocol. SMT can interoperate secure routing protocols and provide comprehensive security, by securing the data transmission phase. Meanwhile, SMT's end-to-end robust fault detection can eliminate abuse of the route maintenance operation of such protocols, preventing adversaries from hiding or reporting erroneous route error messages. Moreover, SMT does not require prolonged observation periods to characterize misbehaving nodes as adversaries, and it is not prone to "blackmail" attacks by adversaries disseminating false misbehavior reports. Finally, SMT does not impose the overhead and the resultant delay of message exchange with all nodes along a faulty route. The performance of such a fault-detection scheme is not presented in the literature; however, its overhead may be justified in the special case that two nodes are unable to communicate for long periods of time. Then, a detection algorithm could be initiated, so that the source discards only links incident to the adversaries, rather than the entire nonoperational route. The system requirements and operational conditions under which such an approach can be practical is part of our future research.

VII. Conclusion

We presented and analyzed the SMT and SSP protocols for secure data communication in ad hoc networks. The two protocols are widely applicable, as they provide lightweight end-to-end security services, and operate without knowledge of the trustworthiness of individual network nodes. They are highly effective, achieving highly reliable, low-delay, and low-jitter communication even in highly adverse settings. SMT can support real-time communication, providing nearly constant delay and jitter, delivering 93% of the messages with no retransmissions, even when 50% of the network nodes disrupt the data transmission. Even with a small number of routes at hand, SMT can deliver, for example, more than 98% of the messages with limited retransmissions, when 30% of the network nodes are adversaries.

SMT and SSP are versatile, as they automatically adapt their operation to resource constrained environments, as well as application requirements. In fact, our protocols span a large space of solutions, offering the flexibility to tradeoff overhead for enhanced fault-tolerance and reliability, or tradeoff delay and delay variability for low overhead. For example, SSP can be equally reliable to SMT, while imposing less than one third of SMT's network overhead. At the same time, they are robust, as their adaptation cannot be abused by adversaries, and resilient to arbitrary data transmission disruptions. Finally, components of SMT and SSP, such as the fault detection or the path survival estimation, could be applicable in other types of networks (e.g., wireline), and for different communication patterns (e.g., multicast).

Overall, the security and fault-tolerance of the data communication are paramount in the inherently insecure and unreliable ad hoc networking environments. This is true for both civilian systems vulnerable to malicious and selfish users and rogue network devices, as well as tactical systems that operate in hostile environments. We believe that protocols such as SMT and SSP, in conjunction with secure routing protocols such as SRP [1], [17], SLSP [5], or QoS-SRP [7], can be the catalyst for the wide deployment of ad hoc network based applications.

APPENDIX

We assume that: 1) there is a finite number of (α, β) pairs; 2) a new path is discovered up to L times if the previously used one is deemed failed; 3) at most D packets are transmitted in total during the protocol session; and 4) path reliabilities take values from a finite set $\{q_i\}$, with $q_i \in [0,1)$. Then, we define the following.

1) The set of *acts* A, as the set of all unordered (α, β)-pairs. We denote the ith pair as *act* α_i.
2) The set of states S, as the set of ordered L-tuples $s_j = (q_1^j, \ldots, q_L^j)$ of path reliabilities.
3) The set of outcomes O, as the set of three-tuples $(\mathrm{TX}, \mathrm{RX}, \mathrm{DR})_{(i,j)}$ of the numbers of *transmitted* (TX) and *received* (RX) packets, and *discovered* (DR) routes for each s_j and α_i.
4) The protocol profit per delivered packet P, the protocol cost per transmitted packet C_1, and the route discovery cost C_2.
5) The *utility function* $u : O \rightarrow R$, for each s_j and α_i

$$u_{\alpha_i}(s_j) = f(\mathrm{RX}, \mathrm{TX}, \mathrm{DR}, P, C_1, C_2). \tag{7}$$

For each state s_j, let α_i^s be the act with the best outcome, which yields the highest utility $u_{\alpha_i^s}(s)$. For each act α_i, we calculate the regret with respect to the utility of the best outcome

$$\mathrm{regret}_u(\alpha_i, s) = u_{\alpha_i^s}(s) - u_{\alpha_i}(s). \tag{8}$$

The maximum regret, over all possible states, if act α_i were performed, is

$$\mathrm{regret}_u(\alpha_i) = \max_{s \in S} \mathrm{regret}_u(\alpha_i, s). \tag{9}$$

We can now derive an order of preference for all acts using the *MiniMax Regret rule*, with an act being more preferable if and only if its maximum regret is lower

$$\alpha_i \geq \alpha_i' \quad \text{iff} \quad \mathrm{regret}_u(\alpha_i) \leq \mathrm{regret}_u(\alpha_i'). \tag{10}$$

We assume that u increases as the number of delivered packets across a path increases, and decreases as the number of transmitted packets and route discoveries increase. We consider the reception of a packet significantly more valuable than the cost of a packet transmission, i.e., $P > C_1$, and the cost of a route discovery higher than the benefit from a packet reception $(C_2 > P)$. For $\mathrm{RX} \leq \mathrm{TX} \leq D$ and $\mathrm{DR} \leq L$, the utility function has the form

$$u_{\alpha_i}(s_j) = \mathrm{RX} \times P - \mathrm{TX} \times C_1 - \mathrm{DR} \times C_2. \tag{11}$$

Since at most D packets are transmitted, the utility is bounded, $-(D \times C_1 + L \times C_2) < u_{\alpha_i}(s_j) < D \times P$, taking negative values when no or very few packets are delivered across the DR paths.

We first consider the selection of α. Independently of the failure pattern, the lower α is, the more failed transmissions are required before the route is deemed failed. Then, all states s_j with $q_1^j, \ldots, q_k^j$ taking low values, for $k < L$, will yield lower utility and, thus, higher regret than those states with relatively more reliable paths utilized first. For $\alpha \rightarrow 0$ and some unreliable route, with $p_i = 0$, $\mathrm{RX} = 0$. The state with the least utility (and, thus, maximum regret) will be the one at which the protocol successively utilizes routes with $q_l^j = 0$, causing $\mathrm{RX} = 0$, $\mathrm{TX} = D$, $\mathrm{DR} = L$. Thus, values for α close to zero must be avoided.

On the other hand, $\alpha \rightarrow r_s^{\max}$ implies that the protocol will tend to discard a route even though it is highly reliable. Now, the regret will be high if q_l^j is high but *not* close to 1, because the protocol will then be driven to discard all routes. The number of delivered packets across each route can be much fewer than the number of packets delivered $(\leq Dq_l^j)$ had the protocol never discarded the route. For example, $q_i = 0.9$ can imply 100 failures occurring after 900 successful transmissions $(u = 900P - 901C_1 - C_2)$, or loss of one every ten packets $(u = 9P - 10C_1 - C_2)$, with two orders of magnitude lower utility in the latter case. Thus, α should not approach $r_s^{\max}$.

The selection of values for α is dominant over the selection of values for β, when α is close to r_s^{thr} or $r_s^{\max}$, since reinstatement of the path rating is irrelevant when the path is never discarded, and it has only limited impact, if any, when almost always the path is discarded independently of its prior history. The selection of β becomes more significant when α takes values in the area between the two extremes. Using $\beta \rightarrow r_s^{\mathrm{thr}}$ implies that a highly reliable route will not be reinstated after one or more failures. If the route is discarded, delivered packets can be fewer than the packets delivered if the route were not discarded. As a result, $\beta \rightarrow r_s^{\mathrm{thr}}$ should not be used. The above discussion leads to the Theorem in Section IV-B.

REFERENCES

[1] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. SCS CNDS*, San Antonio, TX, Jan. 27–31, 2002, pp. 193–204.

[2] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. ACM WiSe*, Atlanta, GA, Sep. 2002, pp. 1–10.

[3] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proc. 8th ACM MobiCom*, Atlanta, GA, Sep. 2002, pp. 12–23.

[4] K. Sanzgiri *et al.*, "A secure routing protocol for ad hoc networks," in *Proc. ICNP*, Nov. 2002, pp. 78–87.

[5] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proc. IEEE CS Workshop on Security and Assurance in ad hoc Netw.*, Orlando, FL, Jan. 2003, pp. 379–383.

[6] Y. Hu, A. Perrig, and D. B. Johnson, "Secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–190, Jul. 2003.

[7] P. Papadimitratos and Z. J. Haas, "Secure QoS-aware route discovery in ad hoc networks," in *Proc. 2005 IEEE Sarnoff Symp.*, Princeton, NJ, Apr. 2005, pp. 176–179.

[8] ——, "Secure on-demand distance-vector routing in ad hoc networks," in *Proc. 2005 IEEE Sarnoff Symp.*, Princeton, NJ, Apr. 2005, pp. 168–171.

[9] S. Kent and R. Atkinson, "Security architecture for the Internet protocol," IETF, RFC 2401, Nov. 1998.

[10] ——, "IP authentication header," IETF, RFC 2402, Nov. 1998.

[11] R. Stewart *et al.*, "Stream control transmission protocol," IETF, RFC 2960, Oct. 2000.

[12] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proc. 3rd ACM MobiHoc*, Lausanne, Switzerland, Jun. 2002, pp. 1–11.

[13] A. Tsirigos and Z. J. Haas, "Analysis of multipath routing, Part 1: The effect on the packet delivery ratio," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138–146, Jan. 2004.

[14] ——, "Analysis of multipath routing, Part 2: Mitigation of the effects of frequently changing network topologies," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 500–511, Mar. 2004.

[15] P. Papadimitratos and Z. J. Haas, "Secure message transmission in mobile ad hoc networks," *Elsevier Ad Hoc Netw. J.*, vol. 1, no. 1, pp. 193–209, Jul. 2003.

[16] ——, "Secure message transmission in mobile ad hoc networks," in *Proc. ACM WiSe 2003*, San Diego, CA, Sep. 2003, pp. 41–50.
[17] P. Papadimitratos, "Secure and fault-tolerant communication in mobile ad hoc networks," Ph.D. dissertation, Cornell Univ., Ithaca, NY, Jan. 2005.
[18] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999. IEEE Standard 802.11.
[19] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.
[20] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Netw. Mag.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.
[21] P. Papadimitratos and Z. J. Haas, "Secure communication in adverse mobile ad hoc networks," in *Ad Hoc Wireless Networking*, D.-Z. Du, Ed. Norwell, MA: Kluwer, Nov. 2003.
[22] ——, Correctness and specification for secure routing in ad hoc networks, under review.
[23] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *J. ACM*, vol. 36, no. 2, pp. 335–348, Apr. 1989.
[24] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-hashing for message authentication, IETF, RFC 2104, Feb. 1997.
[25] R. K. Ahuja, T. L. Magnati, and J. B. Olin, *Network Flows*. Upper Saddle River, NJ: Prentice-Hall, 1993.
[26] M. D. Resnick, *Choices: An Introduction to Decision Theory*. Minneapolis, MN: Univ. Minnesota Press, 1987.
[27] R. Elandt-Johnson and N. L. Johnson, *Survival Models and Data Analysis*. New York: Wiley, 1990.
[28] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling*. New York: Wiley, 2003.
[29] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. New York: Wiley, 1968, vol. 1.
[30] R. E. Barlow and K. D. Heidtmann, "Computing k-out-of-n system reliability," *IEEE Trans. Reliab.*, vol. R-33, pp. 322–323, 1984.
[31] A. M. Rushdi, "Utilization of symmetric switching functions in the computation of k-out-of-n system reliability," *Microelectron. Reliab.*, vol. 26, no. 5, pp. 973–987, 1986.
[32] ——, "Comment on: An efficient nonrecursive algorithm for computing the reliability of k-out-of-n systems," *IEEE Trans. Reliab.*, vol. R-40, no. 1, pp. 60–61, 1991.
[33] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proc. 3rd ACM MobiHoc*, Lausanne, Switzerland, Jun. 2002, pp. 80–91.
[34] J. Broch *et al.*, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. 4th ACM MobiCom*, Dallas, TX, Oct. 1998, pp. 85–97.
[35] J. Yoon, M. Liu, and B. Noble, "Random-waypoint mobility considered harmful," in *Proc. of IEEE INFOCOM* , San Francisco, CA, Apr. 2003, pp. 1312–1321.
[36] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF, RFC 3561, Jul. 2002.
[37] D. B. Johnson *et al.*, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Internet Draft, draft-ietf-manet-dsr-09.txt, Apr. 2003.
[38] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *ACM/IEEE Trans. Netw.*, vol. 9, no. 4, pp. 427–438, Aug. 2001.
[39] T. H. Clausen *et al.*, "Securing the OLSR protocol," in *Proc. IFIP Med-Hoc-Net. 2003*, Jun. 2003, pp. 25–35.
[40] T. H. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF, RFC 3626, Oct. 2003.
[41] B. Awerbuch *et al.*, "An on-demand secure routing protocol resilient to Byzantine failures," in *Proc. ACM WiSe 2002*, Atlanta, GA, Sep. 2002, pp. 21–31.
[42] S. Marti *et al.*, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th MobiCom*, Boston, MA, Aug. 2000, pp. 255–265.
[43] S. Buchegger and J. Y. LeBoudec, "Performance evaluation of the CONFIDANT protocol," in *Proc. 3rd ACM MobiHoc*, Lausanne, Switzerland, Jun. 2002, pp. 226–236.
[44] N. Ben Salem *et al.*, "A charging and rewarding scheme for packet forwarding," in *Proc. 4th ACM MobiHoc*, Annapolis, MD, Jun. 2003, pp. 13–24.
[45] P. Macciardi and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation," in *Proc. 6th IFIP CMS Conf.*, , Sep. 2002, pp. 102–121.
[46] A. Bestavros and G. Kim, "TCP-Boston: A fragmentation-tolerant TCP protocol for ATM networks," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 1210–1217.

Panagiotis Papadimitratos (M'99) received the Ph.D. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2005.

He joined the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, as a Research Associate. He is the author of more than 20 technical papers. His research is concerned with networking protocols, network security, ad hoc and sensor networks, and mobile computing.

Zygmunt J. Haas (S'84–M'88–SM'90) received the B.Sc. and M.Sc. degrees in electrical engineering, and the Ph.D. degree from Stanford University, Stanford, CA, in 1979, 1985, and 1998, respectively.

He then joined AT&T Bell Laboratories, Holmdel, NJ, in the Network Research Department. There he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 until July 1995, he worked for the AT&T Wireless Center of Excellence, Whippany, NJ, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. As of August 1995, he joined the faculty of the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY. He is an author of numerous technical papers and holds 15 patents in the fields of high-speed networking, wireless networks, and optical switching. His interests include mobile and wireless communication and networks, personal communication service, and high-speed communication and protocols.

Dr. Haas is a voting member of the Association for Computing Machinery (ACM). He serves as Editor of several journals and magazines, including the IEEE TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the *IEEE Communications Magazine*, and the *ACM/Kluwer Wireless Networks Journal*. He has been a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Special Issues on "Gigabit Networks," "Mobile Computing Networks," and "Ad Hoc Networks." He has organized several workshops, delivered tutorials at major IEEE and ACM conferences. He is Chair of the IEEE Technical Committee on Personal Communications.