JANUS: A Framework for Scalable and Secure Routing in Hybrid Wireless Networks

Bogdan Carbunar, *Member, IEEE,* Ioannis Ioannidis, and Cristina Nita-Rotaru, *Member, IEEE*

***Abstract*— Hybrid networks consisting of cellular and Wi-Fi networks were proposed as a high-throughput architecture for cellular services. In such networks, devices equipped with cellular and Wi-Fi network cards access Internet services through the cellular base station. The Wi-Fi interface is used to provide a better service to clients that are far away from the base station, via multihop, ad hoc paths. The modified trust model of hybrid networks generates a set of new security challenges as clients rely on intermediate nodes to participate effectively in the resource reservation process and data forwarding.**

In this paper we introduce JANUS, a framework for scalable, secure and efficient routing for hybrid cellular and Wi-Fi networks. JANUS uses a scalable routing algorithm with multiple channel access, for improved network throughput. In addition, it provides protection against selfish nodes through a secure crediting protocol and protection against malicious nodes through secure route establishment and data forwarding mechanisms. We evaluate JANUS experimentally and show that its performance is 85% of the optimum algorithm, improving with a factor greater than 50% over previous work. We evaluate the security overhead of JANUS against two type of attacks: less aggressive, but sufficient for some applications, selfish attacks, and purely malicious attacks.

***Index Terms*— C.2.0 General: Security and protection. C.2.1 Network Architecture and Design: Wireless communication C.2.2 Network Protocols: Routing protocols**

I. Introduction

Cellular services have developed in the last few years from simple telephony to more sophisticated applications. Internet access and mobile multimedia services are available to numerous clients using laptops or PDAs, via special network cards, or mobile phones enabled to act as wireless modems. However, the availability of 3G services depends on the transmission power of the base station. The use of cellular services to their full potential is impeded by the sharp decrease of the achievable cellular rate as a device moves away from the base station. In addition, in metropolitan environments, where 3G services are expected to be mostly in demand, signal propagation can be extremely erratic, even for low-bandwidth transmissions. Addressing these problems by upgrading current cellular networks may be unappealing to service providers, because of financial reasons and restrictive regulations.

An alternative to overcome the limitations of the cellular services relying only on the base station was proposed in [1]. In this architecture the cellular network is augmented with Wi-Fi wireless devices organized into an ad hoc network around the existing cellular one. Nodes that have a satisfactory downlink rate from the base station use other nodes in their proximity as relayers. Such an architecture requires nodes to be equipped with two interfaces: one for cellular communication and one for Wi-Fi communication, such as 802.11a/b/g [2]–[4]. We refer to such a network as a *hybrid wireless network*.

Although hybrid wireless networks have not been deployed by industry yet, major industry cellphone companies such as Nokia or Motorola have released several handsets (e.g. Nokia 6136 and Motorola A910 (Martinique)) that have integrated cellular and ad hoc interfaces. We believe the presence of such devices is an incentive to study the security implications of the deployment of such services. Specifically, while providing opportunities for better service to clients far away from the base station, a hybrid wireless network introduces a significant change in the trust and communication model. In the case of a cellular network, the base station represents a centralized point of trust performing the authentication of all clients, and communicating directly with each client without any intermediaries. In the case of a hybrid network, the trust is no longer centralized, as data flows from the base station to the clients via untrusted intermediate relayers. An intermediate relayer may exhibit selfish behavior by avoiding to cooperate in the process of forwarding data to protect its own resources. A relayer may also exhibit malicious behavior by disrupting the functionality of the network, without looking for an immediate benefit for its actions.

Beyond the security concerns raised by the new trust paradigm, a routing protocol has to be scalable and efficient, with respect to both the throughput it offers and the communication and computational overhead it imposes. Clients that resort to multihop paths to achieve an increased downlink bandwidth from the base station have to derive significant gains from this solution compared to using their cellular link. At the same time, as participating devices have limited resources, clients that act as relayers should not be unduly burdened by the protocol or they will refuse to cooperate.

A. Our Contribution

In this paper we propose JANUS [1], a framework that provides scalable, secure and efficient routing for hybrid wireless networks. JANUS consists of four components: an efficient routing algorithm, a crediting protocol providing

[1] Janus, the Roman god who was guardian of portals and patron of beginnings and endings had his head always shown with two faces: one in the front and one at the back. As Janus, all the hosts in the network have two ”faces”, a cellular and a Wi-Fi communication ”face”.

protection against selfish nodes, and two security components providing resilience against malicious nodes. The first security component protects the path reservation and control messages, while the second achieves an efficient detection and isolation of attacks against data forwarding.

Our routing algorithm, DST, identifies and establishes low overhead multihop paths that are close to optimal. DST achieves this by using a dynamic, spanning tree of the network, rooted at the base station. Maintaining the spanning tree can be costly, demanding as many as $O(n)$ operations per update, where n is the number of clients; instead, DST requires only $O(\log n)$ operations per update, converging lazily to a maximum spanning tree. As a result, DST is extremely efficient when deployed in densely populated areas. Moreover, the routing core is flexible, so that it can be adapted for low network densities, where simpler implementations with higher asymptotical complexities may be more efficient in practice. DST uses a multi-channel MAC scheme to allow concurrent flows and obtain increased throughput.

JANUS uses a crediting protocol to motivate clients to commit their resources to forward data for other participants and prevent selfish hosts from adding false relayers.

For critical applications required to operate correctly under stronger adversarial models, JANUS provides mechanisms to deter destructive malicious attacks that can impact the path establishment and data forwarding services. Our solution takes advantage of the existence of the low-bandwidth cellular secure direct communication between each host and the base station to transmit critical information and efficiently identify and isolate problematic nodes.

We experimentally compare the throughput available to hosts running JANUS with the optimum protocol, a global knowledge Bellman-Ford. While Bellman-Ford imposes heavy traffic on the network, JANUS achieves consistently more than 85% of the throughput of Bellman-Ford. We also show the benefits that can be obtained by taking advantage of multiple channels, even when using a conservative model for interference. We evaluate the computation overhead required by the cryptographic primitives employed by JANUS in order to add and manage client traffic, using Linux phones that are currently on the market. Our simulations show that the average per-host cryptographic computation overhead for maintaining client traffic due to topological changes is under 0.05%.

*R*oadmap: We define the network and security models we assume in Section II. We introduce our framework along with a description of possible attacks and defenses in Section III. We present the experimental analysis in Section IV and overview related work in Section V. Finally, we summarize our conclusions and future work in Section VI.

II. System Model

In this section, we present a detailed description of our system model, regarding both communication and the expected adversarial behavior.

A. Communication and Service Model

The network consists of several mobile hosts and an always available cellular base station. All hosts are within the coverage range of the base station. Each host has a unique identifier [2] and an account registered with the base station. The base station provides Internet access to each registered host, for as long as the host has credit in its account. The content accessed by host clients originates at the base station.

Mobile hosts can communicate with other hosts, using 802.11-compliant (b/a/g) wireless cards. Like many well-known multi-hop wireless routing protocols [5]–[8], our protocol assumes bi-directional links. This assumption is also made by MAC protocols such as 802.11. Uni-directional links can be detected and avoided using techniques such as the ones described in [8]. Hosts use a multi-channel MAC scheme [9] requiring only a single transceiver per host. Such a scheme provides support for multiple non-overlapping channels (3 for 802.11b [3] and 12 for 802.11a [2]).

Interference may occur on the Wi-Fi links. Similar with [10], we model the interference generated by a link transmission as the set of hosts situated in the transmission range of the endpoints of the link. We assume that a mobile host can adjust its data transmission rate. We assume that the cellular channels do not overlap with any of the Wi-Fi ones. A host can support simultaneous cellular and ad hoc communications.

We refer to cellular links from the base station to a mobile host as *forward links* and to cellular links from a mobile host to the base station as *reverse links*. We refer to 802.11 links between any two mobile nodes as *ad hoc links*. Each link $\mathtt{e}$ has a constant weight $\mathtt{w}(\mathtt{e})$ equal to its data capacity. Each node advertises its bandwidth to the base station. A node may decide to use ad hoc links instead of the cellular links when the bandwidth advertised by hosts through the ad hoc links is better than the cellular rate.

B. Security Assumptions

The base station is trusted by each host in the network. The base station authenticates every host via the cellular communication link then establishes a secure cellular channel with each host. Thus, the cellular communication is protected from outside adversaries against eavesdropping, modification of existing packets, and injections of counterfeit packets.

Hosts that can not be authenticated by the base station do not participate in the network and are not trusted. Any host on the path between the base station and a client, although authenticated, may not behave correctly. Attackers can be just selfish – trying to obtain free service and protect their own resources – or malicious – trying to disrupt the function of the network, without consideration for their own resources. An intermediate host can exhibit such behavior either alone or in collusion with other hosts.

A public-key infrastructure is assumed to exist for operations such as signature generation and verification, and shared key establishment, used to protect the ad hoc communication. The base station acts as a Certificate Authority (CA). Each node maintains a cache of nodes and their corresponding public key certificates. Every time a node discovers a new

[2] The unique identifier could be the International Mobile Equipment Identity (IMEI) number of GSM technology.

neighbor it requests its public key certificate and stores the pair in its cache.

We use $\mathtt{E}_{\mathtt{K}}(\mathtt{M})$ to denote encryption of message M with key K, $\mathtt{S}_{\mathtt{X}}(\mathtt{M})$ to denote signature on message M by host X and $\mathtt{HMAC}_{\mathtt{K}}(\mathtt{M})$ to denote the keyed hash of message M using key K.

C. Attacker Model

We assume attackers target well behaved hosts, by manipulating control or data messages. Attackers considered in our work can exhibit both selfish and malicious behavior. Selfish attackers try to dishonestly improve their situation, either by saving or acquiring more resources than they are entitled to receive. They do not have destructive goals and their impact is primarily on the data forwarding service. Malicious attackers however can mount either passive (eavesdropping) or active attacks on both control and data messages. Passive attacks consist of cryptanalysis of forwarded or overheard packets and attempt to reveal encryption keys and data of other hosts. Active attacks involve modification of existing traffic or injection of counterfeit packets. We also assume more powerful, collaborative attacks, either through collusion between existing attackers or corruption and control of multiple hosts.

We do not consider attacks that target client privacy. Network participants, including the base station, may use traffic information in order to build statistics and user profiles. We note that similar with the work in [11], pseudonyms [12] can be used to protect client privacy. We also assume that additional mechanisms are used to provide the confidentiality and integrity of application generated data.

We consider threats occurring only at the network layer in the ISO/OSI model. Attacks occurring at lower layers (e.g. physical layer, MAC) are orthogonal to our work and we do not consider them. Solutions proposed to address attacks at lower levels will benefit any routing algorithm, including our work, JANUS. We refer the reader to Section III-B for a detailed description of the attacks we consider.

III. JANUS Description

In this section we present JANUS, our framework for scalable and secure routing in hybrid wireless networks. We first introduce DST, the core routing algorithm. We then describe in detail potential attacks targeting both control traffic and data traffic and present our proposed defenses consisting of a secure path reservation and a secure data forwarding mechanism. Finally, we describe a crediting and payment mechanism that ensures cooperation in a less adversarial environment.

A. DST Overview

The goal of our routing algorithm is to select for each host, a path providing the highest throughput from the base station. Each host periodically probes its neighbors for their current throughput and selects the neighbor providing the highest value. Such a host is called the *parent*. The period of the neighbor probing is called *refresh rate*. The set of all parents maintained by the hosts represents a *routing tree* encoding the best throughput from the base station to any host. As the link throughput and network topology change, this tree is an approximation of the tree providing best throughput. The accuracy of the tree depends on the refresh rate: a smaller rate ensures a better approximation, but incurs higher overhead. The refresh rate depends only on the volatility of the network - if the network is relatively stable, the refresh rate can be low, without resulting in throughput degradation.

A probed host may need to traverse $\mathtt{O}(\mathtt{n})$ hosts up to the base station, to evaluate its current throughput, n being the total number of hosts. While for small n this is not a problem, for dense, metropolitan areas, the number of messages produced can increase congestion. Our algorithm maintains the routing tree with $\mathtt{O}(\log \mathtt{n})$ number of messages per update, by using a *topology tree* [13] structure.

Path reservation: When a host needs to download data, it contacts its parent in the routing tree, which in turn will contact its parent and so on, until the message reaches the base station. Each contacted host must first locally verify the availability of the resources requested by the client. It then appends its identity to the message received from its child and forwards it to its parent. At the completion of the path reservation process, the base station knows the client host name, the information requested by it, the path to that client, and the bandwidth available on that path.

Data forwarding: Using the information about the established path, the base station can send the client the data required for download at the available rate. Note that data messages will be relayed by intermediate nodes before reaching the client.

Tree management: Four operations are used to manage the routing tree: Cut, Link, Update and Mincost. Cut splits a tree into two subtrees by removing an edge between two vertices. Link joins two subtrees by adding an edge between two vertices, each in a different subtree. Mincost returns the weight of the minimum weight edge on the path from a vertex to the root of the tree. Finally, Update modifies the weight of each edge on the path from a vertex to the root of the tree. A naive implementation has a $\mathtt{O}(\mathtt{n})$ time complexity per operation (Cut, Link, Update, Mincost), where n is the number of mobile hosts. We lower the complexity of these operations to $\mathtt{O}(\log \mathtt{n})$ by using the aforementioned topology trees and the algorithms described in [13]. The routing and topology trees can be stored at the base station in which case the time complexity to perform the four operations represents computational cost. In the case of a distributed implementation of the routing and topology trees, the "time complexity" becomes "communication complexity". Specifically, it represents the total number of base-station-to-host and host-to-host messages needed to perform the tree operations.

We assume that the base station stores and maintains the routing and topology trees. Since a cellular base station has to keep information for every host that is logged in its cell, storing the topology tree does not impose an unrealistic overhead. For instance, in a GSM network, for each host the base station stores the host's IMEI number (15 digits long), a session key (128 bits if AES [14] is used), the parent's IMEI number and information about the link between the host and its parent (two floating point numbers). Effectively, the base station acts as an oracle that answers queries from the hosts on a dynamic, rooted routing tree. The use of topology trees guarantees that such an oracle is efficient, but in principle any implementation of the four operations can serve as an oracle.

Refresh rate selection: Since in dynamic networks routes may quickly become stale, a parent must be reevaluated at constant intervals to adapt to topology changes. We considered two strategies. The first is more aggressive, keeping the parent pointer as close to the optimal as possible with the available information. The strategy requires a node to probe all its neighbors every k seconds. The traffic generated is $O(d \log n)$, where d is the number of neighbors. The second, less expensive, strategy is to probe only the parent every k seconds and cut it only if the rate falls below a threshold. To keep the number of messages per time unit at a scalable level, we modulate k with the size of the network. As the network becomes more dense, the refresh rate should decrease. We provide more details about how we selected the refresh rate in our experiments in Section IV.

Use of multiple channels: Our model of the residual capacity of links due to the addition of flows and the interference they introduce is conservative and may reduce the network throughput. To compensate, we take advantage of the multi-channel capabilities of the 802.11b and 802.11a wireless standards that allows scheduling interfering transmissions to occur simultaneously as long as they use non-overlapping frequencies. Selecting the transmission channel for a new flow is done by a simple traversal of the path. For each traversed link, DST reserves the first locally available channel (see Figure 1(c)). Then, it contacts all the hosts whose potential transmissions interfere with the link (hosts adjacent to the link's endpoints) and reserves the chosen channel. If a link of an interfering host is left without any available channels, its residual capacity becomes zero. This process can be performed using a dedicated control channel on which all idle hosts listen. Moreover, interfering hosts are notified of the reserved channel also on the reserved channel, in order to discover existing, potentially interfering, communications taking place on the same channel. Figure 1(c) shows an example of this approach, where the ad hoc links supporting the newly added flow of A reserve channel 1 and subsequently, channel 1 becomes unavailable for transmissions on links interfering with the flow.

B. Attacks Description

In this section we describe specific attacks we are concerned with in this work and that can affect the different internal components of the DST routing protocol described above and the multi-hop data forwarding. Numerous attacks can be generated by the lack of authentication, integrity and freshness. Not all attacks can be prevented by providing authentication and integrity. For example, in the case of inside (malicious) attackers actions that can not be prevented by authentication and integrity are lying, interfering with data forwarding or resource consumption. We focus on a selective, but representative set of attacks. Attacks targeting the data forwarding service we consider are *freeloading* caused by selfish attackers and *selective data forwarding* caused by malicious attackers.

Algorithm 1 Host's view of the path reservation and data forwarding phases.

```
Object implementation MobileHost;
  BS : BaseStation;
  inQ : InputQueue;
  Id, ssn : integer; #Host and session identifiers
  rate, throld : real; #rate and threshold
  $K_{shr}$ : string; #key shared with BS
  $K_{pub}^{BS}$ : string; #BS's public key
  Operation pathReserve()
    p := new String(INIT, Id, ssn + +, rate, throld, fn);
    pkt := new Packet($E_{K_{shr}}(p)$);
    sendToBS(pkt);
    guard inQ.first.type = SGN do
      verify($K_{pub}^{BS}$, sgn := inQ.first.sgn);
      if correct(sgn, pkt) = false then
        p := new String(ERR, Id, $Id_k$);
        sendToBS(new Packet(ERR, p));
      else
        p := new String(ADDF, Id, ssn, sgn, hmac($K_{shr}$, Id));
        pkt := new Packet(ADDF, p);
        sendToParent(pkt);
    od
  end
  Operation main()
    guard inQ.first.type = ADDF do
      pkt := inQ.first;
      sgn := verify($K_{pub}^{BS}$, pkt.sgn);
      if correct(sgn, pkt) = false then
        p := new String(ERR, Id, $Id_k$);
        sendToBS(new Packet(ERR, p));
      fi
      store(pkt);
      if capacity(parent) < sgn.rate then
        p := new String(LOW, Id, parent,
               capacity(parent));
        sendToBS(new Packet(LOW, p));
        guard inQ.first.type = SGN do
          pkt.sgn := inQ.first.sgn;
        od
      fi
      append(pkt, Id);
      pkt.hmac := hmac($K_{shr}$, pkt.hmac|Id);
      sendToParent(pkt);
    od
    guard inQ.first.type = FLOW do
      pkt := inQ.first;
      if pkt.id = Id then
        if checkHmac(pkt) = true then
          h := hmac($K_{shr}$, ACK|Id|pkt.i + 1|pkt.info);
          sendToBS(newPacket(ACK, Id, h));
      else sendToHost(next(pkt.id), pkt);
    fi
  end
```

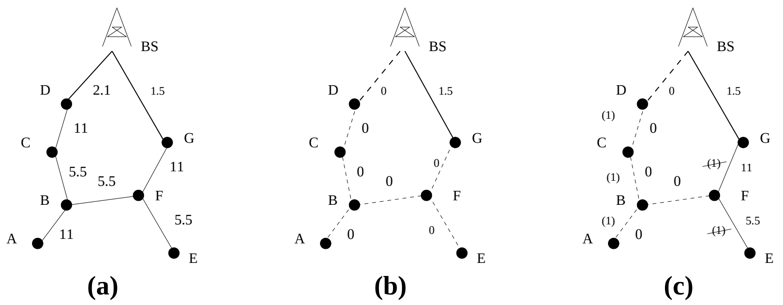


Fig. 1. (a) Example of hybrid network, where labels on the right-hand side of links represent link residual capacities.(b) Residual network of (a) after A adds a flow on links (BS, D),(D, C), (C, B) and (B, A). Due to interference, not only links adjacent to this path but also links of hosts adjacent to this path are blocked.(c) Same scenario as in (b), only using the multi-channel capability of Wi-Fi standards. Labels on the left-hand side of links represent channel assignments. Links of hosts adjacent to the flow path, i.e. (G, F) retain their capacity, but cannot use the channel chosen by A's flow due to interference.

Attacks targeting the routing protocol we consider are: *rate inflation*, *tunneling*, *ghost requests* and *path scrambling*.

Rate inflation: A malicious host can advertise a larger bandwidth to the base station than it can provide. This can be done by inflating the bandwidth of its forward cellular link or of its ad hoc links. A host M that advertises an inflated throughput is able to negatively affect not only adjacent hosts, but also hosts that have M as an ancestor in the routing tree.

Tunneling: Two non-adjacent malicious hosts collude by advertising an excellent bandwidth for the link between them. Their communication can be encrypted and sent through a path of mobile hosts established in ad hoc manner, or simply through the base station. This attack can be also viewed as a particular case of the rate inflation attack performed by a set of colluding nodes, or an instance of a *wormhole* [15] attack.

Ghost requests: This attack is a form of denial of service that results in resource consumption. Malicious hosts waste network resources and generate arbitrary amounts of useless traffic in the network, by making ghost requests for data they do not intend to use. Since each host has a direct cellular link with the trusted base station, malicious colluding hosts cannot prevent other hosts from receiving cellular service. However, they can prevent the clients from taking advantage of the better service provided by the Wi-Fi links and decrease the honest clients' trust in the benefits of the system.

Path scrambling: During path reservation, malicious hosts between the client and the base station can add and remove hosts from the path, or change their order. This attack causes packets to be lost, while allowing the attackers to frame other hosts. Moreover, by manipulating the path, a host can make sure it is selected on a particular path or on many paths, thus controlling a significant part of the network.

Freeloading: A host can try to avoid sharing the cost or responsibility involved in participating in the hybrid wireless network. The intention of a freeloader is just to obtain free service and not to deliberately prevent other nodes from obtaining service. This attack is an attack against the data service. When a payment mechanism or credit scheme is in place, freeloading can additionally target this mechanism, instead of data itself.

Selective data forwarding: A malicious host can selectively drop packets received, instead of forwarding them towards their intended destination. This attack can be devastating, particularly when the attackers are strategically placed, and can potentially drop the throughput of a host to zero, while also diminishing the trust of honest clients in the system.

Algorithm 2 Base station's view of the path reservation and data forwarding phases. All the direct communication between a host and BS is done through the secure cellular links.

```
Object implementation BaseStation;
Operation main()
    guard inQ.first.type = INIT do
        intId := inQ.first.id;
        pkt := decrypt(inQ.first.message, $K_{Id}$);
        store(pkt);
        pkt := sign(Id, pkt.id, pkt.ssn, $K^{BS}_{priv}$);
        sendToHost(Id, new Packet(SGN, pkt));
    od
    guard inQ.first.type = LOW do
        pkt := inQ.first :
        p := signNewRate(pkt);
        sendToHost(pkt.id, new Packet(SGN, p));
    od
    guard inQ.first.type = ADDF do
        pkt := inQ.first;
        if checkHmac(pkt) = true then
            dest := pkt[pkt.size];
            Id := pkt.id;  #pkt.id is client
            String info := $E_{K^{Id_A}_{shr}}$(retrieve($fn_{Id_A}$));
            break(info, n);
            for i := 1 to n do
                h := hmac($K^{Id}_{shr}$, FLOW|Id|i|info[i]);
                p := new String(FLOW, Id, i, info[i], h);
                sendToHost(dest, new Packet(FLOW, p));
            od
        else detectFaultyLink
        fi
    od
end
```

C. Secure Path Reservation

In this section we show how to augment the path reservation protocol presented in Section III-A with security mechanisms that provide protection against ghost requests and path scrambling attacks. We note that our mechanisms do not prevent the formation of tunnels. However, a tunneling attack has no value in itself. The ultimate goal for the attackers is to draw data through them which will allow them to perform traffic analysis or selective data forwarding. Although we do not directly address tunnel formation, we detect such tunnels in the case when the ultimate goal of the attack is selective data forwarding. We discuss this aspect in Section III-E.

Algorithms 1 and 2 present the pseudocode of our secure path reservation using an Orca [16] like syntax. Orca is a parallel programming language for distributed systems, that provides elegant constructions for expressing reactive behavior, such as *guards*. Operations can consist of one or more guards with syntax

guard `expression` **do** `statementSeq` **od** [3].

TABLE I
MESSAGE DESCRIPTION

Type	Description
INIT	Initialize a session
SGN	Signed message from the Base Station
ERR	Error message
LOW	The capacity of the parent is lower than the acceptable threshold
FLOW	Information for establishing a flow
ADDF	Message sent to relayers to notify them they are on a flow
ACK	Acknowledgment sent to the Base Station

The set of message types used are presented in Table I. The protocol works as follows. Host A initiates the path reservation phase with operation `pathReserve`, by contacting the base station through its secure reverse cellular link, (lines 9-11), with a message of type INIT, containing A's identity, $\mathtt{Id_A}$, a session identifier $\mathtt{ssn_A}$, A's throughput, $\mathtt{rate_A}$, a threshold throughput that A considers acceptable, $\mathtt{thr_A}$, and an identifier of the information needed from the base station, `fn`, all encrypted with the secret session key shared by A with the base station. The base station responds by sending A a signed SGN message certifying A's throughput for the given session (lines 54-59). The message is sent through the secure forward cellular link between BS and A. When A receives the message (line 12), it contacts its parent, with a message of type ADDF with the following structure

$$\mathtt{ADDF}, \mathtt{Id_A}, \mathtt{ssn_A}, \mathtt{S_{BS}}(\mathtt{Id_A}, \mathtt{ssn_A}, \mathtt{rate_A}), \mathtt{HMAC}_{\mathtt{K_A}}(\mathtt{Id_A}).$$

When a host N receives an ADDF message (line 24),
$\mathtt{ADDF}, \mathtt{Id_A}, \mathtt{ssn_A}, \mathtt{S_{BS}}(\mathtt{Id_A}, \mathtt{ssn_A}, \mathtt{rate_A}), \mathtt{HMAC_k},\ \mathtt{Id_1},\ \ldots, \mathtt{Id_k}$, where $\mathtt{Id_1}..,\mathtt{Id_k}$ are the identities of all the intermediate hosts from A to N and $\mathtt{HMAC_k}$ is an onion HMAC of all these hosts as reported by $\mathtt{Id_k}$, it checks whether the encoded path is consistent with the topology of N's neighborhood and if the nodes did not process an ADDF message with a smaller or equal $\mathtt{ssn_A}$. If the two checks verify and N has a path to the base station that can accommodate A's request, it forwards the message to its parent, adding its identifier, $\mathtt{Id_N}$ and replacing $\mathtt{HMAC_k}$ with $\mathtt{HMAC_{k+1}} = \mathtt{HMAC}_{\mathtt{K_N}}(\mathtt{HMAC_k})$. The path reservation process ends when the base station receives the ADDF message initiated by A (line 66).

Ghost requests defense: Our defense relies on the INIT/SGN step that each host has to perform with the base station at the beginning of the path reservation phase. The base station authenticates the subsequent ADDF messages, by sending the client host a signed message, that the client host has to use when contacting its parent with an ADDF message. The impact of a denial of service attack is localized by our protocol to the ad hoc neighbors of a malicious host, as invalid ADDF messages cannot be transmitted over more than one link (the immediate neighbors, with the base station's help, detect these messages and stop them). The processing of ADDF packets still consumes resources needed to verify the base station's signature. Hosts can decide to ignore ADDF packets received from repeatedly misbehaving neighbors.

Path scrambling defense: At the end of the path reservation protocol, the base station uses the identities of the hosts from the ADDF packet and their keys shared with the base station to verify the correctness of the HMAC received in the same packet. If an intermediate host tampered with the ADDF message the two values will not coincide. The base station then performs a binary search, in order to retrieve a link that has a malicious host as an endpoint. Let the path received by BS be $\mathtt{P_1}, .., \mathtt{P_k}$. We use $\mathtt{H_i}$ to denote the HMAC value produced by host $\mathtt{P_i}$. The base station finds the median of the path, $\mathtt{m} = (\mathtt{k}+1)/2$ and queries $\mathtt{P_m}$ through the forward link for its local view. The local view consists of the partial path $\mathtt{P_1}, \mathtt{P_2}, .., \mathtt{P_{m-1}}$ and the $\mathtt{H_{m-1}}$ value, received by $\mathtt{P_m}$ from its child, $\mathtt{P_{m-1}}$. If the path $\mathtt{P_1}, \mathtt{P_2}, .., \mathtt{P_{m-1}}$ is consistent with the first half of the initial path $\mathtt{P_1}, .., \mathtt{P_k}$, the base station checks $\mathtt{H_{m-1}}$ against the received path, using the identities of the hosts listed on the path and their keys used to compute the HMAC values.

The search continues on the interval whose left endpoint has a correct HMAC and whose right endpoint has an incorrect value. The search ends when two consecutive hosts on the path received by BS give different results on the HMAC check. If the two hosts are neighbors, the link is considered malicious.

D. Secure Tree Management

Each of the four tree management operations (cut, link, update, mincost) can be exploited by an adversary. For example, intermediate hosts can modify the content of a message signaling a `cut(v)` operation to arbitrarily cut edges in the routing tree. To prevent this attack, the `cut(v)` request to the base station is encrypted using the key shared by the requester with the base station. The `cut` operation is not otherwise a hazardous operation from a security standpoint, since a host is free to reject an unsatisfactory parent.

A malicious node can try to manipulate the `mincost(v)` operation by providing incorrect answers when queried by a host that wants to select a parent in the tree. Since the oracle is assumed to reside in the trusted base station, we envision two

[3] `expression` is a boolean expression and `statementSeq` is a sequence of statements. The operation containing guards blocks until one or more guards are true. Then one of those guards is randomly chosen and its statements are executed atomically.

possible implementations for the mincost operation. In the first solution, a host that needs to find a new parent contacts the base station, providing its list of neighbors and the throughput of the corresponding edges. The base station retrieves the mincost of each neighbor and returns the identity of the one providing the client host with the highest throughput. In the second solution, the base station periodically sends each host a signed and timestamped certificate containing the host's mincost value. When a host needs a new parent, it collects the certificates of all its neighbors, checks their validity and freshness and makes its own local decision.

The link operation can be exploited by an adversarial node to invent links that do not exist or to publish inflated link throughput values. Our solution lies in a secure data forwarding mechanism (see Section III-E) that efficiently discovers faulty, bottleneck links and uses a rating associated with the hosts adjacent to them to reduce the chances of such hosts being chosen as relayers.

The update operation is used whenever a host needs to reserve or release a path of relayers used for downloading information from the base station. It can be exploited by a malicious host to add arbitrary values to the bandwidth of the links of its ancestors in the routing tree. To address this attack, when a path is reserved, the base station performs the update operation using as weight the negative of the minimum bandwidth supported by any relayer. Conversely, the base station performs an update operation with a positive weight only when a download is completed and a path of relayers is released and the resources of the relayers are relinquished. The weight used in this case is equal in absolute value to the negative weight used previously when reserving the path.

E. Secure Data Forwarding

In this section we describe mechanisms designed to ensure secure data forwarding. After receiving an ADDF message and verifying its validity, the base station retrieves the information requested in the INIT message and encrypts it with the symmetric key shared by BS with A (line 71). The encrypted information is split into packets (line 72) and forwarded to the host whose identifier is the last in the ADDF message received by BS (lines 73-77), in messages of type FLOW

$$\mathtt{FLOW}, \mathtt{Id_A}, \mathtt{PID}, \mathtt{PKT_i}, \mathtt{HMAC}_{\mathtt{K_A}}(\mathtt{FLOW}, \mathtt{Id_A}, \mathtt{PID}, \mathtt{PKT_i}),$$

where PID is the packet identifier and $\mathtt{PKT_i}$ is the $\mathtt{i}^{th}$ packet of the flow.

Each host that receives a packet of type FLOW, retrieves from its local database the record corresponding to $\mathtt{Id_A}$, and forwards the packet on the link to the next hop associated with $\mathtt{Id_A}$ (line 49). For each packet received, A ensures its integrity by verifying the HMAC and sends to BS on the secure reverse cellular link, the following ACK packet (lines 46-48)

$$\mathtt{HMAC}_{\mathtt{K_A}}(\mathtt{ACK}, \mathtt{Id_A}, \mathtt{PID}, \mathtt{PKT_i})$$

As in [17], the client can batch the acknowledgments to reduce cellular traffic.

Selective data forwarding defense: During the data forwarding phase, packets sent from the base station to a host can be dropped by malicious hosts trying to interrupt the data flow. JANUS provides an optional mechanism that can be enabled on-demand to detect such events. Our defense against these attacks is based on acknowledgments and the insertion of probes [18]. Similar to [18], we use a threshold on the number of tolerable packet losses, and define a fault to be a packet loss higher than the threshold. Initially, only the client host A needs to send an acknowledgment to the base station. The base station keeps track of the number of packets lost during a certain window of packets. When the number of packets not acknowledged is higher than the acceptable threshold, the base station detects a fault, and initiates a faulty link discovery protocol.

The faulty link discovery protocol consists of selecting a number of intermediate hosts on the path from the base station to the client host A and requesting them to acknowledge future forwarded packets sent by the base station to A. The hosts selected are called *probes*. Each probe P sends the acknowledgments to BS through the probe's secure reverse cellular link. The acknowledgment format is

$$\mathtt{HMAC}_{\mathtt{K_P}}(\mathtt{ACK}, \mathtt{Id_P}, \mathtt{PID}, \mathtt{PKT_i}),$$

where $\mathtt{Id_P}$ is the identity of the probe, and $\mathtt{PKT_i}$ and PID represent the packet acknowledged and its identifier. The selection of the probes is modeled on a binary search of the faulty link. The binary search views the path between the base station and the client A as an interval whose endpoints are BS and A. An interval whose right endpoint does not acknowledge a packet, but whose left endpoint does, is said to be a faulty interval. When a faulty interval is detected, initially the BS to A interval, the interval is divided by selecting as a new probe, the host that is equidistant, in number of hops, to the end points of the interval. The faulty interval division process continues until the faulty interval is a link.

JANUS differs from [18] in several aspects. First, as a side effect of the path reservation phase, the sender BS knows the identities of all the hosts on its path to A. Unlike [18], where the discovery of the intermediaries is broadcast-based, our path reservation protocol requires only unicast on the routing tree path between the base station and A. Second, in [18], the probes are selected by the originator of the traffic, by sending messages on the multihop ad hoc path from itself to each probe. The acknowledgments sent by each probe are also sent on the multihop ad hoc path between the probe and the originator. In contrast, JANUS takes advantage of the dual nature of hybrid networks to separate the control path from the data path, by treating the ad hoc links as the data path and the cellular links as the control path. JANUS transfers all the communication of the probing mechanism via the secure cellular links. The major benefit of this approach is that the packets part of the defense mechanism (probe requests and acknowledgments) are not vulnerable to attacks since they do not travel on the untrusted multihop paths. This reduces the complexity of the protocol and the control traffic.

In summary, the secure data forwarding protocol ensures that the base station detects a link $(\mathtt{u}, \mathtt{v})$, where v is the parent of u, responsible for dropping more packets than advertised.

Once such a link is detected, the base station adjusts its weight. Subsequently, when calling mincost, hosts that use that link may discontinue its use. We defer the discussion of the link weight assignment to Section III-G.

F. Limiting the Impact of Selfish Nodes

In this section, we describe a crediting and billing scheme that deters selfish hosts from receiving fees for services they do not provide or avoiding paying fees for services they have received. Every time a client host sends an INIT message, the client has to be charged by the base station.

The charges are split into a fee for the service provider, a fee for initiating the protocol and the credit to be given to the eventual relayers, if the resulting route is satisfactory to the initiator. The fees for the provider and for initiating the protocol act as a safeguard against ghost request attacks of hosts that request a route without intending to use it. Since the account of the attacker has to be charged, it is easy to detect malicious patterns. The crediting function is of the following form:

$$\begin{aligned} \mathtt{C_A} &= \mathtt{C_{BS}(S)} + \mathtt{C_{INIT}} + \mathtt{C_R(S,r)}, \\ \mathtt{C_R(S,r)} &= \mathtt{G(S)} \cdot \mathtt{F(r)}, \\ \mathtt{F(r)} &= \mathtt{c},\ \mathtt{if}\ \mathtt{r} \leq 2, \\ \mathtt{F(r)} &= \mathtt{k_2} - \mathtt{k_1} \cdot \mathtt{r},\ \mathtt{if}\ \mathtt{r} > 2 \end{aligned}$$

where $\mathtt{C_A}$ is the total credit taken from the initiator, S is the size of the information downloaded by the initiator from the base station, $\mathtt{C_{BS}}$ is the fee for the provider, as a function of S, $\mathtt{C_{INIT}}$ is the initiating fee, and $\mathtt{C_R}$ is the total credit given to the relayers, which is a function of S and of r, the number of relayers. c, $\mathtt{k_1}$ and $\mathtt{k_2}$ are parameters. The function G(S) denotes the participation of S to $\mathtt{C_R}$. The impact of F(r) on the credit the relayers receive can be scaled using G(S). The only part that restricts the form of $\mathtt{C_A}$ is F(r). The amount of credit that each relayer gets is $\mathtt{C_R(S,r)/r}$.

JANUS can be used with any crediting scheme, but, using an appropriate crediting function can prevent selfish hosts from adding spurious relayers and performing freeloading attacks. In the following we propose an instance of F(r), by determining appropriate values for parameters c, $\mathtt{k_1}$ and $\mathtt{k_2}$, that provides counterincentives to freeloading.

Freeloading defense: A way to prevent hosts from adding bogus relayers to a relayer path is to make sure that the sum of credit colluding hosts receive does not exceed their benefit when behaving honestly. We show how the structure of our crediting function $\mathtt{C_A}$ and the structure of F(r), which captures the dependence of $\mathtt{C_A}$ on the number of relayers, guarantees this property.

In the following, we will assume that there are r honest relayers, m true, but colluding, relayers and l false relayers, added by the colluding ones. We also assume that the amount of credit given to the $\mathtt{m+l}$ malicious hosts is divided equally among the m colluding relayers. This is the worst case scenario, as the l false relayers are assumed to be hosts taken over and the credit they receive can be funneled to the m colluding hosts. The credit each host receives without adding false relayers is $\mathtt{F(r+m)/(r+m)}$, while the credit each colluding host receives when adding l false relayers is $\mathtt{(l+m) \cdot F(r+m+l)/m \cdot (r+m+l)}$. We can prove that there are values of c and $\mathtt{k_1}$ and $\mathtt{k_2}$ such that the credit each colluding host receives decreases for $\mathtt{l} > 0$. In the following, T is the maximum number of relaying hops allowed.

Theorem 1: There are $\mathtt{k_1}$, $\mathtt{k_2}$ and c such that

$$\frac{\mathtt{F(r+m)}}{\mathtt{r+m}} \geq \frac{\mathtt{l+m}}{\mathtt{m}} \cdot \frac{\mathtt{F(r+m+l)}}{\mathtt{r+m+l}}, \mathtt{T} \leq 18.$$

Proof:

If $\mathtt{r+m} > 2$, the inequality is equivalent to

$$\frac{\mathtt{k_2 - k_1 \cdot (r+m)}}{\mathtt{r+m}} \geq \frac{\mathtt{[k_2 - k_1 \cdot (r+l+m)] \cdot (m+l)}}{\mathtt{(r+l+m) \cdot m}}.$$

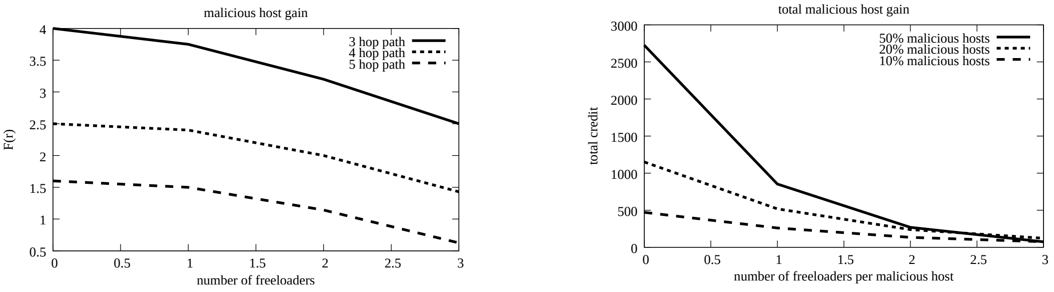
For this to hold for every possible configuration where $\mathtt{r+m} > 2$, it is sufficient to have $\mathtt{k_2} \leq 18\mathtt{k_1}$. When $\mathtt{r+m} \leq 2$, it is necessarily $\mathtt{r} = 0$ and $\mathtt{m} = 2$, otherwise there cannot be an attack. Then the theorem inequality becomes $\mathtt{c} \geq \mathtt{k_2 - k_1 \cdot (2+1)}$. Combining the latter with the condition that $\mathtt{c} > 0$, so that the crediting function is always positive, we get $\mathtt{c} > \max\{0, \mathtt{k_2} - 3 \cdot \mathtt{k_1}\}$.

■

It is easy to see that there are appropriate $\mathtt{k_1}$, $\mathtt{k_2}$, c such that the above inequalities hold. For example, a solution is $\mathtt{k_1} = 2$, $\mathtt{k_2} = 18$ and $\mathtt{c} = 16$. For this parameter values, Figure 2(a) shows that when two malicious hosts add fake relayers and split the credit given to the fake relayers, their profit is significantly smaller. For instance, for a path of three relayers, when two of them add three fake relayers and split the 5 host credits, each malicious relayer receives 62% of the credit it would receive if it behaved honestly.

Moreover, Figure 2(b) shows simulation results in a network of 300 hosts, each moving at a speed of maximum 9m/s, for 100s. 10% (30) of the hosts concurrently support a data flow from the base station. We consider three scenarios, of different concentrations of malicious hosts. That is, we consider a scenario where a host chooses to be malicious with a 50% chance, a scenario where a host has a 20% to be malicious and one where a host is malicious with 10% probability. We experiment with malicious hosts adding 1, 2 and 3 bogus relayers and compare it with the case where they behave honestly (no bogus relayers). We consistently found that for each additional bogus relayer added, the total credit earned by malicious hosts is more than halved. For instance, when each of the roughly 20% malicious relayers adds a single bogus relayer, their total credit gain is 45% of the credit gained if they behaved honestly. We also make the observation that for concentrations of 50% and 10% malicious relayers, each adding 3 bogus relayers, the total credit gain is roughly the same, even though if they behaved honestly, the 50% population would gain more than 5 times more credit than the 10% population. This is because in the 50% malicious population, relayer paths have on average more malicious relayers, thus their total length (true + bogus relayers) increases, leading to a decrease in credit gain.

By updating the behavior of hosts running JANUS to consider not only throughput but also path length when choosing parent hosts, we can further eliminate freeloading. Hosts adding freeloaders would not only receive less credit



(a)

(b)

Fig. 2. (a) Malicious host gain in a scenario with two malicious hosts, adding between 0 and 3 freeloaders and splitting the total profits. Values are plotted for three types of paths, composed of 3, 4 and 5 real hosts. The function $F(r)$ with parameters $\mathtt{k}_1 = 2$, $\mathtt{k}_2 = 18$ and $\mathtt{c} = 16$ is used to determine the per host credit. The use of this crediting function forces honest behavior to be the most profitable. (b) Simulation results in a network of 300 hosts, of which 30 concurrently support a data flow. Three scenarios are depicted, where a host is malicious with a 1/2, 1/5 or 1/10 probability. The plot shows the sum of credit gain of all malicious hosts when each malicious host adds 0,1,2 or 3 fake relayers.

for routing but would also be avoided by other hosts. In the above example, when the two malicious hosts add three freeloaders, the single honest host receives only 25% of the credit it would receive if everyone behaved honestly. Thus, this host has all the reasons to choose to belong to shorter relayer paths. Since topology trees can easily be extended to encode for each host not only the available throughput but also its number of ancestors in the routing tree, a host could drop a parent providing an excessively long path to the base station. Furthermore, when a host receives an ADDF message inviting it to be a relayer, the host can add the number of relayers already visited by the ADDF message to its number of routing tree ancestors. If the result exceeds a certain threshold where its profitability would be unacceptable, the host can decline the invitation. This behavior favors shorter relayer paths, thus on the long term isolating freeloader adders.

G. Link Weight Assignment

Hosts need a rating system for links to evaluate the quality of a path. Our routing algorithm, DST, assigns to each link a weight, which corresponds to its expected bandwidth. In settings where hosts are trusted, an approach for evaluating link bandwidth like in [19] is appropriate. In settings where hosts can be malicious, this approach is not sufficient to handle rate inflation attacks (see Section III-B). If hosts can lie about their links' bandwidths without repercussions, all link weights may be set by hosts to the maximum value. Determining the available throughput a node observes from its parent directly involves that node. In the case the node is malicious and lies and no other sources to confirm the answer exist or, such sources exist but a majority of them are malicious and may lie too, it is provable impossible [20] to know if the node is telling the truth or not. As a result, our protocol operates based on links and not individual hosts. Links that fail to deliver the promised bandwidth can be identified, with our protocol described in Section III-E. After an underperforming link is identified, this information must be incorporated into the system, so that in the future, hosts avoid suspicious links.

One solution is to blacklist hosts at the edge of underperforming links. However, it is possible that only one host is malicious. If the published bandwidth of a link exists, but one of the hosts refuses to forward traffic, so that it can both use the link bandwidth for its own traffic, and also receive credit as a relayer, the other host will be unjustly marked as malicious. Also, it is possible that temporary difficulties or malfunctions cause a link to fail. Marking hosts as malicious is too harsh and can thin out the available pool of relayers quickly.

A better solution is to reduce the weight of the link, so that other hosts are aware that this link is potentially untrustworthy. Hosts will choose paths whose links have good weights, not bandwidth. Hosts may still be misled by inflated rates, but quickly the weight of such links will drop. Malicious hosts cannot inflate rates above the upper bound indicated by the standard, so they cannot compensate for the reduction of their link weight. Links that have not misbehaved for a period of time can be rehabilitated by increasing their weight.

The proposed implementation of DST has the advantage that the base station can choose routing paths without having to communicate with hosts, avoiding this way security threats inherent in a distributed implementation. On the other hand, we must make sure that the overhead incurred to the base station is not beyond reason. We propose to maintain link weights by keeping a link rating for every link in the network. Cellular links are assumed trusted, as the base station can measure the actual bandwidth. Only ratings for ad hoc links need to be kept. When a pair of hosts publishes their link bandwidth to the base station, the base station can multiply the link rating with the bandwidth to produce the link weight. Any of the methods suggested in previous work [18] can be used for increasing and reducing link ratings. However, by keeping a rating for each link, the base station has to store $\mathtt{O}(\mathtt{n}^2)$ information, where $\mathtt{n}$ is the number of hosts, whereas it normally stores only $\mathtt{O}(\mathtt{n})$ information.

Alternatively, we can reduce the storage requirements by assigning ratings to hosts, instead of links. Each host is initially assumed to have perfect rating. The rating of a link is calculated by the base station as the product of the rating of the appropriate hosts. Let hosts $\mathtt{A}$ and $\mathtt{B}$ have ratings $\mathtt{r}_\mathtt{A}$ and $\mathtt{r}_\mathtt{B}$ respectively. Let the published, by either $\mathtt{A}$ or $\mathtt{B}$, bandwidth of link $(\mathtt{A}, \mathtt{B})$ be $\mathtt{l}$. The rating of link $(\mathtt{A}, \mathtt{B})$ is $\mathtt{r} = \mathtt{r}_\mathtt{A} \cdot \mathtt{r}_\mathtt{B}$ and its weight is $\mathtt{r} \cdot \mathtt{l}$. Suppose the base station has detected that $(\mathtt{A}, \mathtt{B})$ is underperforming and its rating should be decreased to

$\mathtt{r}'$. Since we do not know which host is responsible for the low performance of the link, it is natural to decrease the rating of both hosts by the same factor. Let $\mathtt{f} = \mathtt{r}'/\mathtt{r}$. The base station will assign to A a new rating $\mathtt{r}'_\mathtt{A} = \mathtt{r}_\mathtt{A}\sqrt{\mathtt{f}}$ and $\mathtt{r}'_\mathtt{B} = \mathtt{r}_\mathtt{B}\sqrt{\mathtt{f}}$ to B. This will have the indirect effect of decreasing the rating of any link incident to either A or B, which is desirable since malicious behavior is related to hosts, not individual links. The case of increasing the rating of a link after it has performed as promised is identical. We note that once the rating of a host drops below a threshold, the base station could mark the host as malicious, or unreliable, and refrain from routing traffic through this host.

The solution described in this section does not prevent rate inflation and tunneling attacks. However, if a malicious host inflates its links, but still outperforms other hosts, it will be used, but the rating of its links will drop gradually to indicate the actual bandwidth. If a malicious host advertises high bandwidths, but other hosts are better relayers, again the ratings will reflect this and the malicious host will be avoided. Using this approach, we can identify links that have one or both ends malicious hosts and contain the damage caused, in the long run. Given that hosts in cellular networks are known to the base station and their identity cannot change, a rating-based system is an effective method for enforcing cooperation.

IV. Simulation Results

In this section we present simulation results demonstrating the increased performance achieved by JANUS as well as the overhead introduced by the security mechanisms.

A. Simulation Environment

In our experiments we model the ad hoc network by using the unit disk graph model, based on the Agere Short Antenna PC Card Extended specification. We use the ARF [21] mechanism to establish the transmission rate of the communication channel between two mobile hosts. We use the top two transmission rates, of 11Mbps for distances under 160m and of 5.5Mbps for distances under 270m.

The hosts are initially deployed uniformly at random in a square of area $2830 \times 2830\mathtt{m}^2$. We model the dependency between the cellular link rates of hosts and their distance from the cellular base station by using the same approach as the one presented in [1]. The base station is positioned at the center of the $2830 \times 2830\mathtt{m}^2$ deployment square and its cellular transmission range is 1920m. According to this model, each host inside the square is covered by the cellular transmission range of the base station.

We model the movement of each host by using the random waypoint model of [22]. In order to allow the average speed and node distribution to stabilize [23], we discard the first 500s of each simulation. Each host chooses uniformly at random a destination point within the $2830 \times 2830\mathtt{m2}$ deployment square and moves towards it with a speed chosen uniformly at random from the [1,MS] interval. MS is the maximum speed and is specified for each experiment. After reaching its destination, each host immediately repeats this process. Thus, all hosts move continuously.

We use a conservative approach and overestimate the effects of ad hoc link interference, by blocking any transmission involving hosts adjacent to the link. Since in practical deployment, a more optimistic approach to interference can be used, the experimental results we present are a lower bound of the expected throughput performance of JANUS when deployed in a real environment.

We consider the optimal throughput to be the one achieved by running the Bellman-Ford algorithm. Instead of computing the shortest path, we compute the maximum throughput path between the base station and all the mobile hosts it covers. In the case of multiple concurrent flows the optimal for $\mathtt{n}$ flows is computed by running Bellman-Ford on the residual network obtained after removing the bandwidth consumed and the interference introduced by the first $\mathtt{n} - 1$ flows.

We scale the refresh rate, $\mathtt{k}$ with $\log \mathtt{n}$, which generates $\mathtt{O}(\mathtt{d})$ messages per time unit for each host. Even at this rate, the scalability of the network is not affected, as probing one's neighbors with a heartbeat broadcast, an operation performed by almost all wireless interfaces, generates $\mathtt{d}$ replies.

In scenarios using multiple channels, one of the channels is reserved as a dedicated control channel.

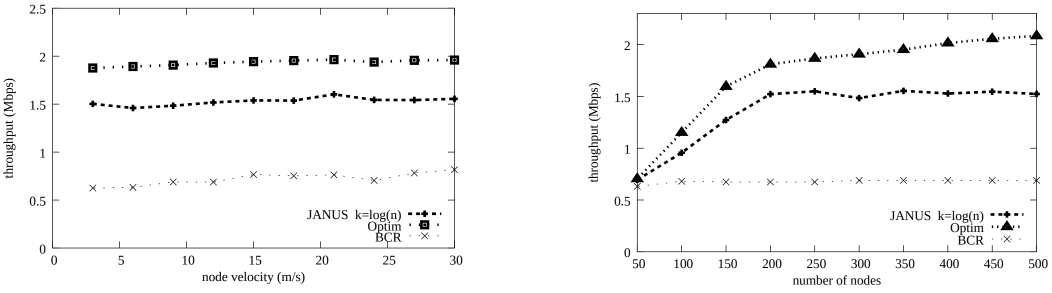
B. Performance Results

In this section we present an experimental analysis of the throughput performance of JANUS with regard to the optimum throughput achievable when a centralized knowledge Bellman-Ford algorithm is executed. We perform each experiment by choosing 5 different initial network configurations and for each such configuration the experiment is run for 100 seconds. In all the graphs presented below, BCR refers to the basic cellular rate, Optim refers to the Bellman-Ford algorithm, while JANUS refers to our protocol.

1) Single flow: The first experiment evaluates the throughput achieved by JANUS under mobility as the speed of the hosts increases from 3 to 30m/s, for a total of $\mathtt{n}$=300 hosts served by the base station. JANUS was configured to use a refresh rate value, $\mathtt{k}$, of $\log \mathtt{n} = 9$s. Figure 3(a) shows that the performance of JANUS follows the trend of the basic cellular rate and is very close, between 75 and 85%, of the optimum throughput achieved by Bellman-Ford. JANUS improves with about 100% over the basic cellular rate.

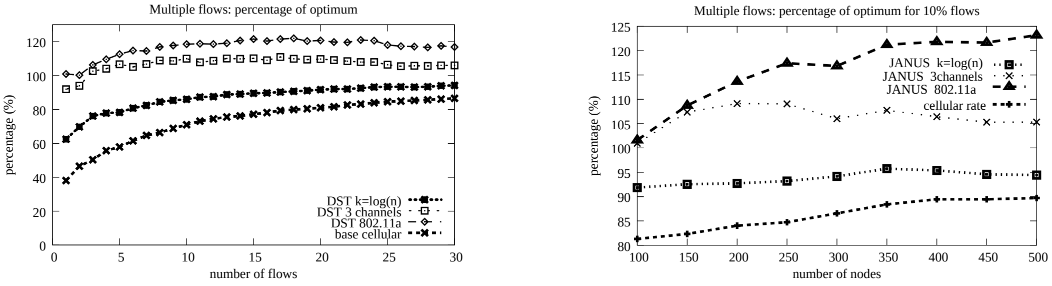
The second experiment explores the dependency between the throughput achieved by JANUS and Bellman-Ford, and the density of hosts served by the base station. In this experiment all the hosts move at a maximum velocity of 9m/s. We measure the evolution of the throughput achieved by the client host when the total number of hosts placed in the deployment square grows from 50 to 500. JANUS is evaluated with a refresh rate, $\mathtt{k} = \log \mathtt{n}$, that for this experiment ranges between 6 and 9s. Figure 3(b) shows the throughput of JANUS compared with the optimal achievable throughput and the basic cellular rate of the client host. It can be observed that JANUS scales very well with the increase in the number of mobile hosts, achieving between 75 and 98% of the optimum.

2) Multiple concurrent flows: We investigate the performance of JANUS when multiple clients support flows simultaneously. We use as baseline the cellular data rate of



(a) (b)

Fig. 3. (a) The throughput achieved for a single flow by JANUS and Optim, for 300 hosts when the maximum velocity of each host increases from 3 to 30m/s. k, the refresh rate of JANUS, is set to $\log \mathtt{n}$, which in this scenario is 9s. (b) The throughput of JANUS and the Optim, for a single flow, as a function of the number of hosts, for a constant maximum speed of 9m/s. The refresh rate of JANUS is $\log \mathtt{n}$, where n is the total number of hosts, thus increasing for this experiment from 6 to 9s.



(a) (b)

Fig. 4. (a) The throughput of JANUS and the basic cellular rate as percentage of the throughput achieved by the Optim when the number of flows grows from 1 to 30 for a network of 300 hosts. We show the results for JANUS run using 802.11b in single channel mode, 802.11b in 3-channel mode and 802.11a using all 12 channels. (b) The throughput of JANUS and the basic cellular rate as percentage of the throughput achieved by the Optim for networks of 50 to 500 hosts, when 10% of the hosts concurrently hold a flow.

the clients and we present the results as a percentage of the optimum throughput achieved by Bellman-Ford. In addition to the performance of JANUS when 802.11b with a single transmission channel is used, we experiment with multi-channel transmissions. We use both 802.11b with its 3 non-overlapping channels and 802.11a providing 12 non-overlapping channels and transmission rates of up to 54Mbps. For our simulations using 802.11a, we experiment with the top two transmission rates, of 54Mbps and 48Mbps.

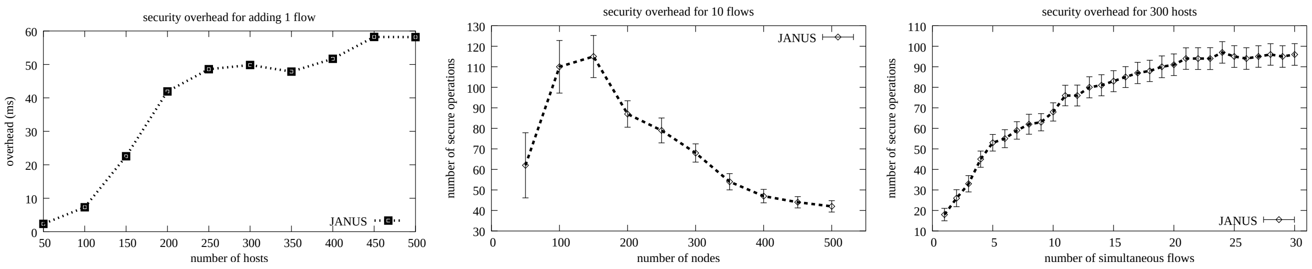
In the first experiment we randomly deploy 300 hosts that continuously move with a maximum speed of 9m/s. We increase the number of simultaneously supported flows from 1 to 30. Figure 4(a) shows the performance of JANUS relative to the optimal total throughput, achieved when all the client hosts run the distributed Bellman-Ford algorithm to find the best downlink path. The performance of JANUS increases to achieve more than 90% of the Optim. In addition, when using the multi-channel capabilities of 802.11b, JANUS exhibits an increase of around 10% over Optim. Even when 30 out of the 300 hosts concurrently support a flow, by using the 3 channels of 802.11b, JANUS achieves a per-flow increase of approximately 200kbps over the basic cellular rate. When JANUS is used in conjunction with 802.11a a 20% increase over Optim is observed.

The second simulation experiments with increasing concentrations of mobile hosts and also of client hosts concurrently supporting flows. In the same square area of $2830 \times 2830\mathtt{m}^2$, we place between 100 and 500 hosts, while also increasing the number of hosts concurrently supporting flows to be 10% of the total number of hosts. Figure 4(b) shows that JANUS constantly achieves more than 90% of Bellman-Ford. The use of the 3 channels of 802.11b allows JANUS to improve the per-flow throughput with 200kbps over the basic cellular rate, or, equivalently, bring a 10% increase over the single channel variant. When JANUS is used in conjunction with the 12 channels of 802.11a, the average per-flow throughput saturates at 1050Kbps, which is 300kbps larger than the basic cellular rate and up to 25% over the Optim. This shows that the employment of multiple channels alleviates the effects of congestion generated at the hosts situated in the vicinity of the base station, by allowing concurrent transmissions on their adjacent hosts.

C. Security Overhead

We evaluate the security overhead imposed by JANUS on hosts in the network. We use as a baseline device an the Motorola E680i GSM phone, equipped with an SD 802.11b card. An E680i is a touchscreen Linux phone with an Intel xScale 300 MHz processor and 50 MB internal memory. We ported OpenSSL 0.9.8 [24] on the E680i and our experiments show that a 1024 bit RSA verification takes on average 2.6ms and a 256 bit SHA1 hash generation takes 38.4μs.

In the first experiment we evaluate the total overhead imposed by performing the security operations required by JANUS, when a single client situated at 1280m from the base station establishes a path to the base station in order to add a flow. Figure 5(a) shows the results of this experiment, performed for configurations containing one base station covering



(a) (b) (c)

Fig. 5. (a) Security overhead, in microseconds, of a client host adding a flow. The client host is placed at 1280m from the base station. The total number of hosts increases from 50 to 500. (b) The average number of messages received by a host during a 1000s run, for configurations containing between 50 and 500 hosts and 10 concurrent flows supported by different client hosts. The intervals represent the corresponding 95% confidence intervals. (c) The average number of messages processed by a host during a 1000s run, for a configuration of 300 hosts, when the number of concurrent flows grows from 1 to 30. The intervals represent the 95% confidence intervals.

between 50 and 500 hosts. Each point on the graph represents an average over 100 different network configurations. For sparse networks (up to 150 nodes), most hosts are either directly connected to the base station or are a small number of hops away from it. Hence the small temporal security overhead imposed by the addition of a flow. For dense networks, the security overhead for adding a flow is under 0.06 seconds.

The following experiment measures the per-host security overhead required for the maintenance of flows concurrently supported by several client hosts. We increase the number of hosts from 50 to 500, while maintaining constant, 10, the number of hosts, simultaneously downloading information from the base station. All hosts move continuously, with a speed of maximum 9m/s, for 1000s. Figure 5(b) shows the average number of messages, with the corresponding 95% confidence intervals, processed by a host during the entire run of the experiment. The maximum number of messages processed by a host is reached for a configuration of 150 hosts. In this case, considering the Motorola E680i phone as the host of choice, on average, a host has to validate a message, taking roughly 2.7 ms, every 8 seconds. For configurations of 500 hosts, the overhead is even smaller, since the cost distributes among many hosts and a host will process a message only once in 20 seconds.

The last experiment measures the per-host security overhead generated by a network configuration of 300 hosts, each moving at a maximum speed of 9m/s, when between 1 and 30 flows are concurrently supported by hosts in the network. Figure 5(c) shows the results of this experiment, run for 1000s. Each point on the graph represents the average number of per-host processed messages during the entire run of the experiment. The per-host number of processed messages increases sublinearly with the increase in the number of simultaneously supported flows and saturates at under 100 messages per host. This is equivalent to processing a message on average once every 10 seconds and results in the acceptable per-host time overhead of 0.027%.

The bulk of the overhead imposed by JANUS on a client host is the symmetric key decryption of packets received from the base station. Our experiments with OpenSSL and 128 bit AES in CBC encryption mode on a E680i phone show that this lightweight host can decrypt 2Mbytes per second. Since this exceeds the rate at which it can receive packets through the 802.11b card, the client host can decrypt packets in a thread running in parallel to the receiving thread, thus substantially reducing the decryption overhead.

V. Related Work

Hybrid wireless networks have been introduced only recently, and research on hybrid routing algorithms has been limited. In this section, we provide an overview of previous research conducted in several areas related with our work: architectures for hybrid wireless networks, ad hoc and hybrid routing and security for ad hoc and cellular wireless networks.

Architectures for Ad Hoc and Hybrid Wireless Networks: The most popular model of wireless networks in the literature is that of the ad hoc architecture [25], [26], [22]. The distributed nature of ad hoc networks limits their scope, as maintaining a connected network over a large area is quite difficult. There have been efforts to integrate infrastructure-based network models with ad hoc components, but most of them assume single-interface devices. In [27], GSM terminals are used to relay information to other terminals to improve coverage. In Opportunity Driven Multiple Access [28], transmission power is conserved by relaying traffic from a CDMA host to the base station through multiple, short hops. In [29], some channels are reserved for forwarding when the fixed channels become congested. In [30], a generic wireless network is considered, where hosts contact a mobile base station for access outside their cell, using only one interface. In [31], a hybrid network using the IEEE 802.11 architecture with both DCF and PCF modes is examined, using only one wireless interface. In [32], multihop paths are used to decrease the number of base stations by increasing their coverage. The overall capacity increases only when two communicating hosts are in the same cell.

Although double-interface architectures are conceptually similar to their single-interface counterparts, they increase the overall capacity by using short-range, high-bandwidth, ephemeral channels to relay traffic and a long-range, low-bandwidth, permanent channel to complete operations like routing and data integrity confirmation or as a last resort in the absence of neighbors. The low-bandwidth channels are not necessarily cellular, but the already existing infrastructure make them attractive options. This architecture has been examined in [1]. In [33], traffic is diverted to neighboring cells to increase throughput. The use of dedicated, stationary relays increases the cost of their solution and limits its utility. A study of local area hybrid networks is presented in [34]. A comprehensive presentation of a rudimentary hybrid network can be found in [35].

Routing in Hybrid Wireless Networks: UCAN [1] provides two routing algorithms that allow clients to find hosts that are willing to forward their traffic. The first algorithm greedily propagates the routing request to a single neighbor, based on the best rate advertised by its neighbors. The second algorithm is based on a restricted depth flooding to find the best proxy host. However, since both algorithms ignore the finite capacities of the ad hoc links, they do not find the highest throughput path.

Fujiwara, Iida and Watanabe [36] provide a unicast routing algorithm that allows mobile hosts to find an alternate path to the base station, when the cellular connection fails. The focus of the work is on reestablishing connectivity to the base station via short paths, in response to emergency situations where direct connections to the base station may become sparse. This is a different goal from JANUS, which attempts to optimize throughput under normal conditions, where all hosts have a direct connection to the base station via the cellular interface.

Security in Wireless Networks: Security in hybrid wireless networks, although necessary for actual deployment, has been left largely unexplored in previous work. UCAN [1] focuses on preventing individual hosts from deleting legitimate hosts or adding non-authorized hosts to the set of relayers that receive credit for forwarding data. It provides protection against isolated, selfish nodes, but it is entirely defenseless when faced with collusion. The work in [37] provides incentives for collaboration in a network where mobile hosts are covered by several access points. The main assumption of the paper is that the hosts are selfish, but not malicious. Moreover, the solution provided does not describe the underlying routing algorithm, but only assumes that there is one and that it is secure. The work in [17] provides incentive to encourage packet forwarding in ad hoc wireless networks. Finally, the work in [38] focuses on preserving the anonymity and privacy of mobile hosts in a network covered by several access points.

The problem of defining compelling methods to make nodes participate in forwarding data was also addressed in the context of traditional ad hoc wireless networks [39] and resulted in designing protocols that provide fair access to the medium. However, most of the research has focused on securing routing protocols and designing key management schemes. The addressed attacks include impersonation and replay and the solutions rely on secure association, symmetric-key based cryptographic mechanisms, or digital signatures [40]–[43]. More sophisticated attacks such as wormhole [15], [44], flood-rushing [45], or arbitrary Byzantine behavior [18], [46], [47] were also considered. Some of the limitations of these works are not considering colluding attackers, and not working correctly in multi-rate networks [46] or sparse networks [47]. We note that in most of the cases, the work focused on securing specific protocols, such as DSDV [48], DSR [49] or AODV [26].

A suite of stealth attacks utilizing routing information manipulation to achieve network partitioning and traffic hijacking is proposed in [50]. These attacks are not effective against JANUS because the base station is always accessible through cellular links, thus can never be removed from the routing information.

The cellular network protocols and standards also include a security component. The core services provided by any of the cellular communication protocols (such as GSM [51], GPRS [52] or UMTS [53]) are authentication of subscribers (clients), providing subscriber identity confidentially, and confidentiality and integrity of both data communication and radio signaling. Clients are authenticated using a challenge-response protocol, based on a key and using algorithms stored on the Subscriber Identity Module (SIM) smart card, obtained when the client subscribes to the service.

VI. Concluding Remarks

In this paper, we have described the challenges of designing a high-throughput, scalable and secure routing mechanism for hybrid wireless networks. We proposed JANUS, a framework that addresses these issues with a fast routing core, and several security components. Our framework is flexible, and can be tailored to obtain optimal performance both in small and large networks. The security mechanisms provide protection against malicious attackers and, for environments with less restrictive security requirements, against selfish attackers. Our experimental results demonstrate that JANUS can provide hybrid wireless networks with much improved bandwidth without significant overhead.

VII. Acknowledgments

This work is supported by National Science Foundation CyberTrust Awards No. 0430276 and 0545949. The views expressed in this research are not endorsed by the National Science Foundation.

References

[1] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu, "Ucan: a unified cellular and ad-hoc network architecture," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 353–367, ACM Press, 2003.

[2] *IEEE Std 802.11a-1999*. http://standards.ieee.org/.

[3] *IEEE Std 802.11b-1999*. http://standards.ieee.org/.

[4] IEEE, *IEEE Std 802.11g-2003*. 2003. http://standards.ieee.org/.

[5] C. Perkins, "Ad-hoc on-demand distance vector routing," 1997.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, (New York, NY, USA), pp. 234–244, ACM Press, 1994.

[7] B. R. Bellur and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," in *INFOCOM*, pp. 178–186, 1999.

[8] P. Jacquet, P. Mhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," in *IEEE INMIC'01, 28-30 December 2001, Lahore, Pakistan*, pp. 62–68, 2001.

[9] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proceedings of MobiHoc*, 2004.

[10] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 9–19, ACM Press, 2004.

[11] M. Raya and J.-P. Hubaux, “The security of vehicular ad hoc networks,” in *SASN ’05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, 2005.

[12] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[13] G. N. Frederickson, “Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees,” *SIAM J. of Comp.*, vol. 26(2), pp. 484–538, 1997.

[14] *Advanced Encryption Standard (AES)*. No. FIPS 197, National Institute for Standards and Technology (NIST), 2001. http://csrc.nist.gov/encryption/aes/.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Packet leashes: A defense against wormhole attacks in wireless ad hoc networks,” in *INFOCOM*, April 2003.

[16] H. E. Bal, R. Bhoedjang, R. Hofman, C. Jacobs, K. Langendoen, T. Ruhl, and M. F. Kaashoek, “Performance evaluation of the orca shared-object system,” *ACM Trans. Comput. Syst.*, vol. 16, no. 1, pp. 1–40, 1998.

[17] N. B. Salem, L. Buttyan, J.-P. Hubaux, and M. Jakobsson, “Node cooperation in hybrid ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 365–376, 2006.

[18] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, “An on-demand secure routing protocol resilient to byzantine failures,” in *ACM Workshop on Wireless Security (WiSe)*, September 2002.

[19] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 134–146, ACM Press, 2003.

[20] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, 1982.

[21] A. Kamerman and L. Monteban, “Wavelan-ii: A high performance wireless lan for the unlicensed band,” *Bell Labs Technical Journal*, 1997.

[22] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, vol. 353 of *Mobile Computing*. Kluwer Academic Publishers, 1996.

[23] J. Yoon, M. Liu, and B. D. Noble, “Random waypoint considered harmful,” in *INFOCOM*, (San Francisco, CA), April 2003.

[24] “OpenSSL: The Open Source toolkit for SSL/TLS.” `http://www.openssl.org/`.

[25] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *MOBICOM*, pp. 85–97, 1998.

[26] C. E. Perkins and E. M. Royer, *Ad hoc Networking*, ch. Ad hoc On-Demand Distance Vector Routing. Addison-Wesley, 2000.

[27] G. Aggelou and R. Tafazolli, “On the relaying capacity of next-generation gsm cellulat networks,” February 2001.

[28] T. Rousse, I. Band, and S. McLaughlin, “Capacity and power investigation of opportunity driven multiple access (odma) networks in tdd-cdma based systems,” 2002.

[29] X. Wu, S.-H. Chan, and B. Mukherjee, “Madf: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure,” in *Proceedings of IEEE WCWN*, 2000.

[30] I. Akyildiz, W. Yen, and B.Yener, “A new hierarchical routing protocol for dynamic multihop wireless networks,” in *Proceedings of IEEE INFOCOM*, 1997.

[31] H.-Y. Hsieh and R. Sivakumar, “On using the ad-hoc network model in wireless packet data networks,” in *Proceedings of ACM MOBIHOC*, 2002.

[32] Y.-D. Lin and Y.-C. Hsu, “Mulithop cellular: A new architecture for wireless communications,” in *INFOCOM*, 2000.

[33] S. De, O. Tonguz, H. Wu, and C. Qiao, “Integrated cellular and ad hoc relay (icar) systems: Pushing the performance limits of conventional wireless networks,” in *Proceedings of HICSS-37*, 2002.

[34] S. Lee, S. Banerjee, and B. Bhattacharjee, “The case for a multi-hop wireless local area network,” in *INFOCOM*, 2004.

[35] M. Miller, W. List, and N. Vaidya, “A hybrid network implementation to extend infrastructure reach,” tech. rep., University of Illinois at Urbana Champaign, January 2003.

[36] T. Fujiwara, N. Iida, and T. Watanabe, “An ad hoc routing protocol in hybrid wireless networks for emergency communications,” in *IEEE ICDCS2004 (WWAN2004)*, pp. 748–754, March 2004.

[37] N. B. Salem, L. Buttyan, J.-P. Hubaux, and M. Jakobsson, “A charging and rewarding scheme for packet forwarding in multi-hop cellular networks,” in *MobiHoc*, 2003.

[38] S. Capkun, J. P. Hubaux, and M. Jakobsson, “Secure and privacy-preserving communication in hybrid ad hoc networks,” Tech. Rep. IC/2004/10, EPFL-IC, January 2004.

[39] P. Kyasanur and N. Vaidya, “Detection and handling of MAC layer misbehavior in wireless networks,” in *International Conference on Dependable Systems and Networks (DSN’03)*, 2003.

[40] P. Papadimitratos and Z. Haas, “Secure routing for mobile ad hoc networks,” in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, pp. 27–31, January 2002.

[41] Y.-C. Hu, D. B. Johnson, and A. Perrig, “SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks,” in *The 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[42] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” in *MOBICOM*, September 2002.

[43] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, “A secure routing protocol for ad hoc networks,” in *10th IEEE International Conference on Network Protocols (ICNP’02)*, November 2002.

[44] L. Hu and D. Evans, “Using directional antennas to prevent wormhole attacks,” in *NDSS 2004*, 2004.

[45] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Rushing attacks and defense in wireless ad hoc network routing protocols,” in *ACM Workshop on Wireless Security (WiSe)*, 2003.

[46] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *MOBICOM*, August 2000.

[47] P. Papadimitratos and Z. Haas, “Secure data transmission in mobile ad hoc networks,” in 2^{nd} *ACM Workshop on Wireless Security (WiSe)*, 2003.

[48] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *ACM SIGCOMM’94*, 1994.

[49] D. B. Johnson, D. A. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. in Ad Hoc Networking*, ch. 5, pp. 139–172. Addison-Wesley, 2001.

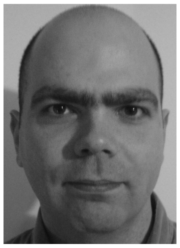
[50] M. Jakobsson, S. Wetzel, and B. Yener, “Stealth attacks on ad hoc wireless networks,” in *Proceedings of VTC*, 2003.

[51] E. Barkan, E. Biham, and N. Keller, “Instant ciphertext-only cryptanalysis of gsm encrypted communication,” in *Proceedings of CRYPTO2003*, pp. 600–616, LNCS, 2003.

[52] “GPRS security threats and solutions.” White Paper by NetScreen Technologies Inc., March 2002.

[53] V. Niemi and K. Nyberg, *UMTS Security*. Wiley Publishers, December 2003.

Bogdan Carbunar is a senior staff researcher at Motorola Labs, where he is developing applications for mobile wireless devices. He received his BS in Computer Science from Politehnica University of Bucharest, Romania in 1999, and a PhD Degree in Computer Science from Purdue University in 2005. His interests include applied cryptography, with applications in electronic payments and secure data outsourcing. He is a member of IEEE.



Ioannis Ioannidis has received his BS in Computer Engineering from University of Patras, Greece in 1999 and his PhD in Computer Science from Purdue University in 2005. His research interests are optimization of backbone IP routing tables, routing algorithms in mesh networks, cryptography.

Cristina Nita-Rotaru is an Assistant Professor in the Computer Science department of the Purdue University. where she leads the Dependable and Secure Distributed Systems Laboratory. She received the BS and MS degrees in Computer Science from Politechnica University of Bucharest, Romania, in 1995 and 1996, and a PhD degree in Computer Science from Johns Hopkins University in 2003. She received the National Science Foundation CAREER award in 2006. Her research interests include secure distributed systems, network security protocols in wired and wireless networks. She is a member of the ACM and IEEE Computer Society.