SECURITY AND COMMUNICATION NETWORKS
*Security Comm. Networks.* 2010; **3**:250–260
Published online 13 January 2010 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.173

# A Dirichlet reputation system in reliable routing of wireless *ad hoc* network

Li Yang[1*,†], Alma Cemerlic[1] and Xiaohui Cui[2]

[1]*Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, U.S.A.*
[2]*Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Applied Software Engineering Research Group, Oak Ridge, TN, U.S.A.*

## Summary

*Ad hoc* networks are very helpful in situations when no fixed network infrastructure is available, such as natural disasters and military conflicts. In such a network, all wireless nodes are equal peers simultaneously serving as both senders and routers for other nodes. Therefore, how to route packets through reliable paths becomes a fundamental problems when behaviors of certain nodes deviate from wireless *ad hoc* routing protocols. We proposed a novel Dirichlet reputation model based on Bayesian inference theory which evaluates reliability of each node in terms of packet delivery. Our system offers a way to predict and select a reliable path through combination of first-hand observation and second-hand reputation reports. We also proposed moving window mechanism which helps to adjust ours responsiveness of our system to changes of node behaviors. We integrated the Dirichlet reputation into routing protocol of wireless *ad hoc* networks. Our extensive simulation indicates that our proposed reputation system can improve good throughput of the network and reduce negative impacts caused by misbehaving nodes. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Dirichlet reputation; reliable routing; *ad hoc* network

## 1. Introduction

A wireless *ad hoc* network is a decentralized wireless local area computer network that allows wireless devices to directly communicate with each other. Because of their decentralized nature, minimal configuration, and ability to be quickly deployed, *ad hoc* networks are especially suitable for emergency situations such as natural disasters or military conflicts.

*Ad hoc* networks are sometimes referred to as a 'self-organized communication system' [1]. They are characterized by the absence of infrastructure and organized according to the peer-to-peer (P2P) principle [1]: all the nodes participating in a network have equivalent responsibilities (they are all peers). Another type of network, an infrastructure mode wireless network, relies on the power of access points to cover a wide communication area and relay data

*Correspondence to: Li Yang, Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, U.S.A.
†E-mail: Li-Yang@utc.edu

Copyright © 2010 John Wiley & Sons, Ltd.

between the devices in the network. Conversely, *ad hoc* networks have no designated routers or access points, routing decisions are made depending on current connection availability, and all nodes in a network have equal responsibility to forward packets for all the other networked nodes. The communication range of an *ad hoc* network depends on the power available in the participating devices. If a device tries to send data to another device not within its communication range, the data must be forwarded through multiple nodes in order to reach its destination [2].

Most wireless *ad hoc* routing protocols are concerned only with maintaining the connectivity among the nodes and assume that participating nodes will cooperate and perform their routing duties for other nodes. It is reasonable to expect users of a self-organized communication system to be concerned primarily about their own benefit, thus cooperation and fairness cannot be guaranteed. Since *ad hoc* networks are completely dependent on willingness of their participants to forward packets for each other, misbehaving nodes diminish quality of service and endanger integrity of network traffic.

Various types of reputation systems have emerged in recent years as an attempt to address these problems. Considering the established security models, the need for the reputation systems is obvious. The traditional information security, commonly defined as 'the preservation of confidentiality, integrity, and availability of information (CIA model)' can be applied on standard wireless networks. However, the CIA model is not adequate for protection against deceitful service providers [3]. Rasmussen and Jansson [3] first described 'soft security mechanisms'. The purpose of these mechanisms is to stimulate ethical behavior and integrity of members in collaborative environments such as *ad hoc* networks, where the ethical norms are not fixed but are rather dynamically defined by its participants. Soft security mechanisms are able to recognize and sanction intolerable behavior and reward members who follow the norms. Reputation systems are classified as a type of soft security mechanisms [3]. When applied to *ad hoc* networks, reputation systems require every node to monitor their peers' behavior. This information is then used to determine which peers can be cooperated with and which should be avoided [1].

In order to improve reliability and security of a wireless *ad hoc* network, we proposed to incorporate a fine-grained reputation system into a wireless *ad hoc* protocol with the aim to protect the network from misbehaving nodes [4]. Similar works [3,5] were proposed to use multinomial Dirichlet distribution in reputation and service reliability evaluation. Our system is able to make a distinction between two types of misbehavior, selfish and malicious and to take an appropriate action upon detecting either type. In this work, we integrate Dirichlet reputation into routing of wireless *ad hoc* networks, and improve our previously proposed reputation system with moving window mechanism. The moving window mechanism makes our system adaptive to behavior change of wireless nodes. Dirichlet reputation system together with moving window mechanism is the major contribution of this paper, which helps to improve routing of wireless *ad hoc* network in terms of packets delivery. We use simulation to show the benefits of the granular approach as opposed to the reputation systems based on the binomial distribution that classify misbehaving nodes as either selfish or malicious. Moreover, moving window mechanism helps to improve performance of wireless routing based on Dirichlet reputation.

## 2. Threat Models and System Assumptions

Based on observations about their forwarding behavior, we classify wireless nodes into three categories: friendly, selfish, and malicious. Different from binomial classification of nodes into good and bad categories, our approach allows for more precise categorization and finer granularity.

Friendly nodes correctly forward packets they receive from other nodes. These packets are delivered to the destination with preserved data integrity. Friendly behavior is the expected behavior in the *ad hoc* networks. Selfish nodes drop packets they receive for forwarding from other nodes, but they expect the other nodes to forward packets that the selfish nodes send. Selfish behavior may be a result of a node's physical properties (loss of battery power, overload with forwarding requests), purposeful attempt to save its own resources (battery and computing resources), or a random failure. This class of misbehaving nodes lowers reliability of the network. Malicious nodes misroute, modify, or inject packets (making them a part of a different data transfer). These nodes are primarily interested in attacking and damaging the network. Malicious nodes lower security and integrity of the network traffic.

For various reasons, a node's behavior can change. For example, if a node starts losing its battery power, it may begin to behave selfishly and drop packets. Even though this node was previously categorized as

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

friendly, our reputation system will react to its recent selfish behavior and reclassify it as selfish.

All nodes which behave in such way that they are degrading security, integrity, or reliability of the *ad hoc* network we define as adversaries. We group the adversaries according to consistency of misbehavior and whether they are selfish or malicious. In this work, we consider the following adversaries' attack models:

- *Model 1*: consistent individual malicious nodes: a malicious node *always injects or modifies* packets when it is in a selected routing path. Our reputation system is able to detect and avoid such nodes.
- *Model 2*: occasional individual malicious nodes: a malicious node *occasionally injects or modifies* packets when it is in a selected routing path. The moving window mechanism allows us to adjust the responsiveness of the fine-grained reputation system to changes in nodes' behavior. If the window size is small, the system will quickly recognize any occasional changes in behavior.
- *Model 3*: consistent individual selfish nodes: a selfish node *always drops* packets when it is in a selected routing path. Our system is able to recognize and avoid such nodes.
- *Model 4*: occasional individual selfish nodes: a selfish node *occasionally drops* packets when it is in a selected routing path. This model is handled in the same manner as Model 2.

We assume that *ad hoc* nodes are capable of bidirectional communication on every link. This assumption means that it is possible to send data from node A to node B or from node B to node A at any point in time. Many wireless Medium Access Control (MAC) layer protocols, including IEEE 802.11, require bidirectional communication for reliable transmission. We also assume that network interfaces on the nodes support promiscuous mode operation. Promiscuous mode 'means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A' [6].

## 3. Related Work

Reputation systems have been used in various domains: E-commerce systems such as eBay [7], Yahoo auctions [8], and P2P storage [9,10]. The works that are close to our context are reputation systems designed for *ad hoc* networks, Confidant [11] and Core [12], and P2P networks [13]. Both *ad hoc* network and P2P networks are distributed and also maintain a statistical representation of the reputation by using tools from game theory. The systems try to counter selfish routing behavior of nodes by forcing nodes to cooperate with each other. More recently reputation systems proposed in the area of *ad hoc* networks employs Bayesian analytics [14]. Reputation-based systems are applied to wireless sensor networks by Ganeriwal *et al*. [15] as well.

Wang *et al*. [16] classify reputation systems that use incentives to encourage cooperation are classified as motivation based. On the contrary, detection-based reputation systems exclude misbehaving nodes from the network soon as their reputation falls below certain threshold. Regardless of the strategy taken against misbehaving nodes, all reputation systems should be fully decentralized and scalable in order to fit the *ad hoc* architecture.

Motivation-based reputation systems do not involve active monitoring of the network, but only work to encourage packet forwarding. A motivation-based solution presented in Ref. [17] involves introduction of a virtual currency called nuglets. Buttyan *et al*. [18] suggest implementation of nuglet counters which are increased for forwarding services and decreased for sending. Detection-based reputation systems such as those implemented in Refs. [11,16,19], and Ref. [12] do not offer incentives but threaten to exclude misbehaving nodes from the network when their reputation falls below certain threshold. Excluded nodes are not allowed to either send or forward packets.

In our work, we consider misbehaving nodes include malicious nodes and selfish nodes. The former one can generate more threats to *ad hoc* network than the latter one. This calls for an ability to distinguish between selfish and malicious nodes. Our system employs Dirichlet distribution, Bayesian approach, and moving windows mechanism to make this distinction.

## 4. A Dirichlet Reputation System

The Dirichlet reputation system is designed to improve reliability and security in wireless *ad hoc* networks. In a fine-grained reputation system, each node stores reputation information for other nodes in the network. This reputation information indicates how well the node behaves as a participant in the wireless *ad hoc* network. Based on the reputation of the intermediate nodes, nodes in an *ad hoc* network are able to select the most reliable and secure path from the data source to the destination. Reputation information indicating that a particular node behaves inadequately will trigger

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

a response from the rest of the network resulting in disadvantages or a punishment for the misbehaving node. The reputation system allows for a different response depending on the severity of the node's bad behavior. This feature is an advantage in comparison to the binomial reputation systems.

The total reputation value, used in routing decisions, is evaluated from both first-hand and second-hand reputation. The first-hand reputation value is based on direct observations in promiscuous mode that node A makes about the behavior of node B. The second-hand reputation value is obtained by sharing the first-hand reputation value with neighbors. All nodes are required to periodically broadcast their first-hand reputation values to their neighborhood. When node A receives a report form its neighbor C about node B's behavior (denoted as $F_{\mathrm{C,B}}$), it will merge the two values, the first-hand reputation $F_{\mathrm{A,B}}$ and the second-hand reputation $F_{\mathrm{C,B}}$, in order to calculate the total reputation value for node B. We only allow propagation of first-hand reputation. Buchegger *et al.* [1] explain that 'passing on information received from others, as opposed to direct observation, turns out to not only offer no gain in reputation accuracy or speed, but also to introduce vulnerabilities by creating a spiral of self-reinforcing information'.

We use a trust mechanism to protect the reputation system from incorrect second-hand reputation reports. The trust value indicates how accurately a node reports reputation information to other nodes. Each node is required to calculate and record trust values for its neighbors when it receives second-hand reputation values from them. If node C is reporting to node A about node B, trust reflects node A's opinion about trustworthiness of the report considering that it came from node C. Since every node regards its knowledge to be the most accurate, it will compare the received information with its own. The new information will only be accepted if it fits within a specified acceptable deviation.

Next we discuss details of each component of the Dirichlet reputation system.

### 4.1. First-Hand Reputation

Each node in the Dirichlet reputation system is responsible for observing the forwarding behavior of its neighbors using the promiscuous mode on its network interface. For example, node A observes the number of correctly forwarded, dropped, or maliciously modified packets from its neighbor, node B. Based on outcomes drawn independently from these observations, node A assumes that the behavior of node B follows a probability of $F_{\mathrm{A,B}}$. We assume that $F_{\mathrm{A,B}}$ follows the Dirichlet distribution. The Dirichlet distribution is the conjugate prior of the parameters of the multinomial distribution. The multinomial distribution is a discrete distribution which gives the probability of choosing a given collection of $m$ items from a set of $n$ items with repetition and the probability of each choice given by $x_1, \ldots, x_n$. The probabilities are the parameters of the multinomial distribution. The probability density of the Dirichlet distribution for variables $x = (x_1, \ldots, x_n)$ with parameters $\alpha = (\alpha_1, \ldots, \alpha_n)$ is defined by $p(x) = \mathrm{Dirichlet}(x|\alpha) = (1/Z(\alpha)) \prod_{i=1}^{n} x_i^{\alpha_i - 1}$ when $x_1, \ldots x_n \geq 0$; $\sum_{i=1}^{n} x_i = 1$ and $\alpha_1, \ldots, \alpha_n > 0$. The parameters $\alpha_i$, as the shape parameters for the probability density function (pdf) of the Dirichlet distribution, can be interpreted as 'prior observation counts' for events governed by $x_i$. The normalization constant $Z(\alpha)$ becomes $Z(\alpha) = \prod_{i=1}^{n} \Gamma(\alpha_i) / \Gamma(\sum_{i=1}^{n} \alpha_i)$.

In Bayesian probability, a class of prior probability distributions $P(\theta)$ is said to be conjugate to a class of probability functions $P(x|\theta)$ if the resulting posterior distributions $P(\theta|x)$ are in the same family as $P(\theta)$. When a conjugate prior is multiplied with the likelihood function, it results in a posterior probability that has the same functional form as the prior, allowing the posterior to be used as a prior in further computations.

Given the Dirichlet prior $P(x_i) = \mathrm{Dir}(x_i|\alpha_{i1}, \ldots, \alpha_{ir})$, the posterior distribution of $\theta_i$ can be computed using the Bayes' theorem as $P(\alpha_i|D) = P(D|\alpha_i)P(\alpha_i)/p(D) = \mathrm{Dir}(\alpha_i|\alpha_{i1} + N_{i1}, \ldots, \alpha_{ir_i} + N_{ir_i})$where $D$ are new data and $N$ describes instances of new data. Starting with the initial state as the prior distribution, the parameters can be updated when new data $D$ is available. We define the reputation value based on the first-hand observations assigned to a node at a time to be equal to the expectation value of the $\mathrm{Dir}(\alpha)$. Given that we are observing three possible events: friendly, selfish, or malicious behavior, the parameters$\alpha_1$, $\alpha_2$, and $\alpha_3$ indicate whether a node is friendly, selfish, or malicious. Let observations of correctly forwarded, dropped, or maliciously modified packets be $x = (x_1, x_2, x_3) \sim \mathrm{Dir}(\alpha)$, and $\alpha_0 = \sum_{i=1}^{3} \alpha_i$, then the expectation value of the distribution is $E[x_i] = \alpha_i/\alpha_0$ [20]. Initially, we choose an optimistic approach and classify every node as friendly. The shape parameters $\alpha_1$, $\alpha_2$, and $\alpha_3$ are thus $\langle 1, 0, 0 \rangle$. Regardless of the initial shape parameters, as new information is collected, the parameters $\alpha_1$, $\alpha_2$, and $\alpha_3$ will converge to approximate the node's true behavior. Algorithms 4.1 is designed to evaluate reputation of a neighbor based on past experience a node had with its neighbor.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

**Algorithm 4.1 EVALUATION OF FIRST-HAND REPUTATION**

```
//calculate in regular intervals
//initially we assume all nodes are friendly
_curRepFriendly = 1;
_curRepSelfish = 0;
_curRepMalicious = 0;

if (_observationWinTotal.length == WINDOW_SIZE)  {
    //categorize collected observations
    for ( i < window_size ) {
        _totalObservedPkts = _totalObservedPkts + _observedWindowTotal[i];
        _friendlyObservedPkts = _friendlyObservedPkts + _observedWinFriendly[i];
        _selfishObservedPkts = _selfishObservedPkts + _observedWinSelfish[i];
        _maliciousObservedPkts = _maliciousObservedPkts + _observedWinMalicious[i];
      }

          //calculate first-hand reputation
          _expFriendly = (_friendlyObservedPkts) / (_totalObservedPkts);
          _expSelfish = (_selfishObservedPkts) / (_totalObservedPkts);
          _expMalicious = (_maliciousObservedPkts) / (_totalObservedPkts);

          //merge first-hand observed with the current reputation value
          _tempFr = curRepFriendly + _expFriendly;
          _tempSe = curRepSelfish + _expSelfish;
          _tempMa = curRepMalicious + _expMalicious;
          _tempTotal = _tempFr + _tempSe + _tempMa;

          //normalize current reputation value <f,s,m>
          curRepFriendly = _tempFr/_tempTotal;
          curRepSelfish = _tempSe/_tempTotal;
          curRepMalicious = _tempMa/_tempTotal;
      }
```

We use information in Table 1 to illustrate how to evaluate reputation of a node based on its packet delivery behavior. A node A is observing behaviors of a node B. Originally, A has no knowledge about B's behavior and is not able to classify B into any category, so shape parameters are set the same (we use 1 here) as $\alpha_1 = \alpha_2 = \alpha_3 = 1$. Each time new data are collected based on observations, $\alpha_1$, $\alpha_2$, and $\alpha_3$ are updated. For example, in the first window, 50 packets are observed and node B correctly forwards 50 ($N1$) packets, drops 0 ($N2$) packets and does not misroute or inject any packet ($N3$=0). The parameters are updated as $\alpha_1 = \alpha_1 + N1$, $\alpha_2 = \alpha_2 + N2$, and $\alpha_3 = \alpha_3 + N3$. If A has observed up to five windows, $\alpha_1$, $\alpha_2$, and $\alpha_3$ are updated to 220, 0, and 30, respectively. The reputation vector of node B is $\langle 0.88, 0, 0.12 \rangle$, where 0.88 (=220/250) is the friendly factor, 0 (=0/250) is the selfish factor, and 0.12 (=30/250) is the malicious factor.

## 4.2. Moving Window Mechanism

Nodes in the Dirichlet reputation system observe their neighbors over equal time intervals or windows. The size of the observation window will affect evaluation results of reputation. If we use a small moving window, the reputation system will be extremely responsive to even minor changes in nodes behavior. In the case of a wireless network this may be undesirable. A wireless network is prone to errors in packet content and occasional packet dropping due to interference or other wireless-medium-related causes. The reputation system could overreact and label a node as misbehaving only because of medium-related issues. If we use a large size of observation window, the system may not be able to immediately detect changes in node's behavior.

We update first-hand reputation based on the most recent observation for the following reasons: it reduces computation complexity, it makes possible to early detect changes in subjects behavior, and it provides the possibility of redemption over time for a node that has been repaired.

Table 1 illustrates a case in which node A is observing node B which was behaving friendly during the first three windows, but then became malicious.

From Table 1 we see that if the size of our moving window is 5, the first-hand reputation of node B

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

Table 1. Illustration of early detection by moving window mechanism.

| Window | Number of observed packets | $\alpha 1$ | $\alpha 2$ | $\alpha 3$ |
|---|---|---|---|---|
| 1 | 50 | 50 | 0 | 0 |
| 2 | 50 | 50 | 0 | 0 |
| 3 | 50 | 40 | 0 | 10 |
| 4 | 50 | 40 | 0 | 10 |
| 5 | 50 | 40 | 0 | 10 |
| From window **0** to **5** | 250 | 220 | 0 | 30 |
| Expectation value of Xi | – | **0.88** | **0** | **0.12** |
| From window **3** to **5** | 150 | 120 | 0 | 30 |
| Expectation value of Xi | – | **0.8** | **0** | **0.2** |

observed by node A is $F^5_{AB} = \langle 0.88, 0, 0.12 \rangle$. If the size is 3, only the three most recent observations are considered and the first-hand reputation is $F^3_{AB} = \langle 0.8, 0, 0.2 \rangle$. We see that $F^3_{AB}$ better reflects the change of behavior in node B, as the malicious component of $F^3_{AB}$ is higher than in $F^5_{AB}$.

Besides using most recent information, our *moving window mechanism* is designed to detect misbehaviors fast and redeem bad reputation slow. If behavior of a node is becoming maliciously, the size of moving window will be reduced by half. With reduced observation window, the reputation evaluation will drop faster; therefore, misbehaving nodes can be detected faster. If behavior of a malicious node is trying to become friendly, the size of moving window will be doubled. With increased observation window, it will take a malicious node longer time to redeem its reputation, which benefits the wireless *ad hoc* network as a group. Algorithm 4.2 shows how to adjust moving windows to evaluate reputation of a node with changing behaviors.

## 4.3. Second-Hand Reputation

If *ad hoc* nodes only relied on their own observations, they would have a localized view of the network and would not be able to make informed routing decisions. Additionally, promiscuous mode monitoring occasionally results in false observations due to packet collisions or other medium-related errors. Because of these limitations, nodes in the Dirichlet reputation system gather as much information as they can before making a decision about their neighbor's behavior. We employ collaborative monitoring to allow nodes to exchange observations with each other. In our Dirichlet reputation system, nodes are required to periodically broadcast first-hand reputation values they collected into their neighborhood, so that other nodes can incorporate this knowledge into their total reputation values. We refer to these reports as second-hand reputation.

In order to detect and avoid false reports, nodes use a deviation test on all reports they receive. Based on the test, nodes decide whether to increase or decrease the trust in the reporter's accuracy, and ultimately whether or not to incorporate the report into their total reputation value. The formula for the deviation test we use is: $|\mathrm{Mode}_{AB}(\beta_1, \beta_2, \beta_3) - \mathrm{Mode}_{CB}(\alpha_1, \alpha_2, \alpha_3)| \leq d$, where the first term is the second-hand reputation, the second term is the current reputation value the node has, and $d$ is a positive constant as the threshold. The Mode is the highest point in the histogram. Buchegger et. al. [21] uses expectation of Beta distribution in their work to perform deviation test. Alternative way to evaluate similarity between first-hand reputation and second-hand reputation is the Euclidean distance. If the first-hand reputation vector is $< \alpha_1, \alpha_2, \alpha_3 >$ and the second-hand reputation vector is $< \beta_1, \beta_2, \beta_3 >$,

**Algorithm 4.2 MOVING WINDOW**

```
//if malicious component increases above the threshold
//the size of the moving window is increased twice during the redemption process
if(currentRepMalicious > MALICIOUS_TRESHOLD)   {
      if (previousRepMalicious <= _expMalicious)  {
            //node is continuing to behave maliciously
            //the size of the moving window is reduced
            window_size = ORIGINAL_WINDOW_SIZE/2;
      }  //end if
      else  {
            //node is starting to behave less maliciously
            window_size = 2 * ORIGINAL_WINDOW_SIZE;
      }  //end else
}  //end if
//node is either redeemed or not malicious
//if the window size has been change, resize it to the original value
else if (window_size > ORIGINAL_WINDOW_SIZE)  {
      window_size = ORIGINAL_WINDOW_SIZE;
}  // end else if
```

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

the deviation test based on Euclidean distance is $\sqrt{(\alpha_1 - \beta_1)^2 + (\alpha_2 - \beta_2)^2 + (\alpha_3 - \beta_3)^2} \leq d$, where $d$ is a positive constant as the threshold. We used 0.3 in our simulation experiments.

### 4.4. Trust of Second-Hand Reputation Report

Each node in the Dirichlet reputation system, in addition to reputation information, stores trust information for its neighbors. Trust indicates how trustworthy the neighbor's reports are. Our trust system uses a similar Bayesian approach used in the Dirichlet reputation system. The difference is that trust is expressed through only two possible instances of behavior: trustworthy and not trustworthy (as opposed to the reputation, where a node can behave friendly, selfishly, or maliciously). Because of this, we choose to use the Beta distribution as a prior. The Beta distribution is in fact a case of the Dirichlet distribution with only two pdf shape parameters. The Beta distribution is conjugate, which means that a posterior probability will have the same functional form as the prior. Therefore, after updating, the trust value still follows the Beta distribution.

We denote the trust which node A has for node B as $T_{AB} \sim \text{Beta}(\gamma, \delta)$, where $\gamma$ stands for trustworthy reports and $\delta$ for not trustworthy. Initially, $\gamma = \delta = 1$, which corresponds to uniform distribution and indicates absence of knowledge. Trust is updated when results of the deviation test are available. If the deviation test holds, the trust for the reporting node is increased by increasing $\gamma = \gamma + 1$. If the deviation test does not hold, the trust is decreased by increasing $\delta = \delta + 1$. Each time new deviation test results are available, trust parameters are updated.

Trust value for node B on node A is calculated as the expectation value of $\text{Beta}(\gamma, \delta)$ as $\omega_{AB} = E(\text{Beta}(\gamma, \delta)) = \gamma/\gamma + \delta$ [15].

### 4.5. Total Reputation Value Based on First-Hand and Second-Hand Reputation

In order to calculate the total reputation value used in routing decisions, an *ad hoc* node merges its own observations (first-hand reputation) with observations reported by other nodes. Every node believes that its prior knowledge about the behavior of its neighbor, node B, is the most accurate. Thus, first-hand reputation information is merged as-is, while second-hand information is discounted by the factor $\omega_{AC}$ expressing the disbelief in the accuracy of the report. The total reputation value node A calculates for node B, given second-hand reputation reported by node C is calculated as: $R_{AB} = F_{AB} + \omega_{AC} F_{CB}$ [21], where $F_{AB}$ is the first-hand reputation, $F_{CB}$ is the second hand reputation, and $\omega_{AC}$ is the trust node A has for node C's reports.

## 5. Integration in Wireless *Ad Hoc* Routing

Most *ad hoc* routing protocols assume that the participating nodes will cooperate and perform their forwarding duties for other nodes. The only task such protocol needs to handle is maintaining connectivity among the nodes. Typically the 'goodness' of a path is calculated based on a cost of each individual node on that path. The cost of each node is assigned using some cost function. The protocols search for the path that has the minimum total cost, i.e., the least-cost path. The simplest example is the shortest-path approach, where each node has the same cost of one. When the routing protocol searches for a path, it will find the one that contains the smallest number of intermediate nodes, since its cost will be the smallest.

If the *ad hoc* routing protocol does not have a mechanism to detect presence and activity of misbehaving nodes, the chosen path may contain selfish and/or malicious nodes. As a consequence, in the presence of non-cooperative nodes, the performance of the network will degrade, until ultimately the network is rendered useless.

The advantage that a reputation system offers to a routing protocol is the ability to classify nodes into good and misbehaving, use the total reputation value as a path cost estimate, and take appropriate actions against the misbehaving nodes in order to protect the security and reliability of the network. Our Dirichlet reputation system further extends this feature by allowing the protocol to classify nodes in one of the three categories – friendly, selfish, or malicious, and treat selfish and malicious nodes separately. By definition, malicious nodes compromise the integrity of the data packets, thus we consider them more dangerous than selfish nodes. While both security and reliability of an *ad hoc* network are important, we think that the damage inflicted by a security breach is more likely to be serious and irreparable than a loss of data packets caused by selfish nodes. Because of this, we suggest that malicious nodes should have a slower redemption time during which they are temporarily isolated by disallowing them to send packets.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

The Dirichlet reputation system is a distributed reputation system, meaning that each node stores a reputation table representing its own view of its environment. In other words, a global view of the reputation for all nodes does not exist. Our design is an appropriate match for the decentralized, infrastructure-less architecture of wireless *ad hoc* networks.

In a wireless routing scheme coupled with the Dirichlet routing protocol, each node assigns cost to other nodes depending on the current total reputation values for those nodes. Since it is more practical to describe cost as one numerical value, we convert the reputation 3-tuple (⟨f,s,m⟩) into a single-valued cost using the following function: $C(f_A, s_A, m_A) = af_A + bs_A + cm_A$, where $a$, $b$, and $c$ are constants such that $1 \leq a < b < c$, and $< f_A, s_A, m_A >$is the total reputation value for node A for which holds that $f_A + s_A + m_A = 1$. The constants $a$, $b$, and $c$ are used to distinguish between the cost of friendly, selfish, and malicious nodes. The constants can be adjusted in such way where nodes with a certain high probability of selfish behavior are considered to have the same cost as malicious nodes, or where a certain amount of malicious behavior is tolerable as it can be attributed to transmission errors. Regardless of the approach, the cost boundaries between friendly and selfish, and selfish and malicious nodes have to be clearly defined so that the routing protocol is able to distinguish among the groups. For instance, we consider selfish nodes to always have greater cost than the friendly nodes (regardless of how small the probability of selfish behavior may be), and similarly, the malicious nodes to always have greater cost than the selfish nodes. We express this relationship as: $1 \leq af_A < bs_A < cm_A$. The upper bound for $cm_A$ is equal to the infinite cost indicating a disconnected link. If one path in the network has a slightly better reputation than other paths, it will most likely be selected by multiple sources as the best path. It is possible that nodes on the path become overloaded by forwarding requests. These nodes will not be able to forward all the packets they receive and will start dropping packets, thereby increasing their selfish component. Eventually, the reputation of the nodes on this path will decrease, and a new best path will be selected. The friendly nodes which were not able to process packets because of their physical limitations and unrealistic expectations will be treated as selfish. Since the routing protocol will keep seeking the least-cost path, this scenario could continue repeating indefinitely. To balance packet load on friendly nodes, we suggest that instead of one best path, the routing protocol selects $m$ disjointed 'best' paths, if such exist. Disjointed means they do not have any intermediate nodes in common. In the set of selected paths, one path will be the best and the inferior paths may have a greater selfish component. It is likely that one of the selected paths is better than the rest. However, we let the source occasionally, with a predetermined (lower) probability, select one of the inferior paths. By doing this, the selfish nodes on the path are given a chance to improve their behavior, and the otherwise friendly nodes on the best path will have a lower chance of becoming overloaded.

As we discussed previously, the moving window mechanism allows us to adjust responsiveness of the Dirichlet reputation system to changes in nodes' behavior. The smaller the size of the moving window, the quicker is the system to detect misbehavior. The moving window allows us to implement a gradual redemption of malicious nodes once they have been isolated. If we were to allow them to quickly recover their reputation, malicious nodes may be tempted to misbehave again. The procedure is the following: once a malicious node reaches the reputation threshold for isolation, we do not allow it to send any packets. We increase the moving window size used for monitoring of this node in order to increase the time needed for the node to recover. In case the node continues to behave maliciously, the moving window size will be readjusted to the original (smaller) size.

## 6. Simulation and Results

In order to evaluate the benefits of the Dirichlet reputation system and moving window mechanism, we have performed four sets of experiments: (1) wireless routing with beta reputation system that classify nodes into good (g) and selfish (s); (2) wireless routing with beta reputation system that classify nodes into good (g) and malicious (m); (3) wireless routing with Dirichlet reputation system that classify nodes into good (g), selfish (s), and malicious (m); and (4) wireless routing with Dirichlet reputation system and moving window mechanism. We will refer to systems in first two experiments as a beta reputation system because it assumes that the nodes' behaviors follow the Beta distribution. We consider two variations of the beta reputation system: a system ⟨g, s⟩ which classifies all misbehaving nodes as selfish, and a system ⟨g, m⟩ which classifies all misbehaving nodes as malicious.

In our simulations, data packets are sent at an average rate of 50 packets/s. Injected packets are assigned to a

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

packet flow from a source to a destination when the flow is routed through a malicious node. We assume that the routing paths are provided and maintained by the underlining routing protocol. We also assume that the reputation information is exchanged in a correct manner regardless of a node's reputation classification. For our simulation cases, we use a constant number of 50 nodes randomly placed on an 800 m × 800 m stage. The diameter of the maximum communication range for each node is 75 m. Every node has at least one edge connecting it to the rest of the network. All edges are bidirectional, meaning that if nodes A and B are neighbors, and A is able to receive packets from node B, node A could instead send packets to node B. We assume that the mobility rate in the *ad hoc* network is low considering the duration of our simulation cases (500 s). The pattern of packet exchange among the nodes is random for each simulation case. The same cases are run on the Dirichlet and the beta reputation systems. During the simulation, packets are sent from a source to a destination every second. All misbehaving nodes belong to one of the two classes – they are either selfish or malicious. For the purpose of the simulation cases, we define selfish nodes as the ones that consistently drop approximately 80% of the packets they are supposed to forward for other nodes. We define malicious nodes as the ones that consistently inject 10 packets per 50 forwarded packets. The total per cent of misbehaving nodes varies from 4% to 30%. We follow the example set in Ref. [12] and limit the population of misbehaving nodes to 30% of the total node population. Michiardi *et al*. claim that this is in most cases unrealistically high ratio of misbehaving nodes, and thus we take it to be sufficient to test the performance of the reputation systems. In each simulation case, the ratio of selfish and malicious nodes is different, but their total number never exceeds 30% of the total size of the node population. We group simulation cases in four main sets according to the total number of misbehaving nodes: 30%, 20%, and up to 10%. We then vary the ratio of selfish and malicious nodes within each set.

In order to evaluate the performance of our model, we consider the following metrics: good throughput and the percentage of injected or modified packets manipulated by malicious nodes. Good throughput is a measurement of the effective aggregate bandwidth of our system. It is the ratio of the bandwidth occupied by the legitimate (not modified or injected) packets that reached their destination and the bandwidth carrying all the packets that reached their destination. We express this mathematically as:

$$\mathrm{GTP_N} = \frac{\left(\frac{P_\mathrm{R}-P_\mathrm{I}}{P_\mathrm{S}}\right)}{\left(\frac{P_\mathrm{R}-P_\mathrm{I}}{P_\mathrm{S}} + \frac{P_\mathrm{I}}{P_\mathrm{S}}\right)}$$

where $\mathrm{GTP_N}$ is the good throughput for a network during some defined time interval, and $P_\mathrm{R}$, $P_\mathrm{I}$, and $P_\mathrm{S}$ are, respectively, the total number of received, injected, and sent packets during this time interval.

Finally, the percentage of injected or modified packets indicates how often malicious nodes receive an opportunity to forward data packets for other nodes. The Dirichlet reputation system distinguishes malicious from selfish behavior and avoids malicious nodes when making routing decisions. Our goal is to decrease the number of injected or modified packets present in the system when compared to the reputation system that classifies nodes as good and bad.

The following is a comparison of the Dirichlet reputation system and a reputation system able to classify behavior of *ad hoc* nodes only as good (g) or bad (either malicious (m) or selfish (s)). We consider two variations of the beta system – the one that treats all nodes as malicious and the other that treats all nodes as selfish. The reputation systems are compared using *good throughput* and the *percentage of injected packets* introduced in the network by malicious nodes.

Figure 1 compares the good throughput in four reputation systems when we have 30% misbehaving nodes: the Dirichlet reputation system with moving windows mechanism, Dirichlet reputation system, the beta reputation system that considers misbehaving nodes as malicious nodes, and the beta reputation system that considers misbehaving nodes as selfish nodes. The results show that the Dirichlet reputation system has overall higher effective throughput than either of the beta reputation system variations. Moreover, the moving window mechanism improves good throughput in the Dirichlet reputation system. Since in the beta reputation systems both selfish and malicious behavior are considered to have equal cost, the beta reputation systems are likely to make a mistake and choose malicious over selfish nodes when estimating a path cost. Such false decisions will compromise the security of the network, which we always consider to be more costly than a possibility of packet loss. In general, we can say that we expect a better effective throughput from the Dirichlet reputation system than the beta reputation systems that treat all misbehaving nodes as either selfish or malicious. This is especially true in the presence of a larger number of malicious nodes.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

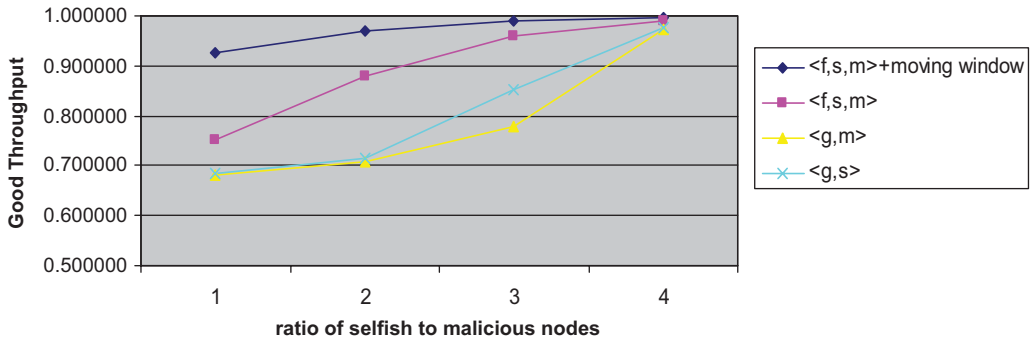


Fig. 1. Comparison of good throughput in four systems (with 30% misbehaving nodes).

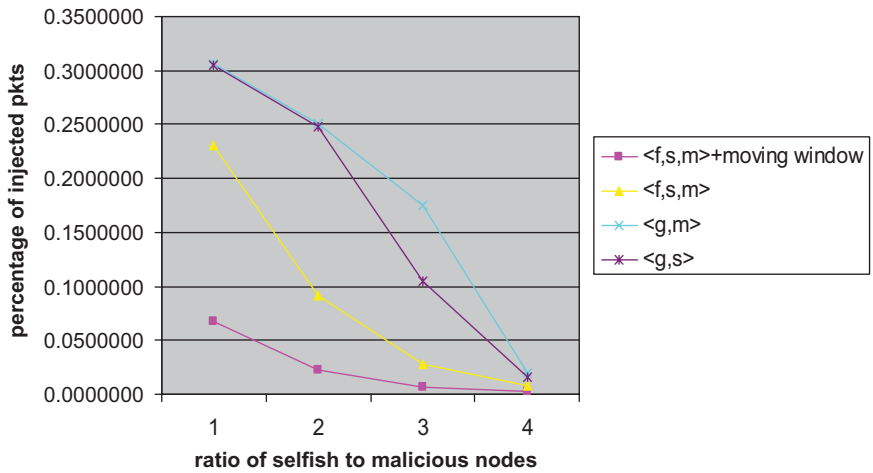


Fig. 2. Comparison of packet rate injected packets in four systems (with 30% misbehaving nodes).

Figure 2 shows a comparison of the four reputation systems with 30% misbehaving nodes. Four systems are compared in terms of the packets manipulated by malicious nodes. We observe that the per cent of injected packets is lower for the Dirichlet reputation system than for either of the beta reputation systems. The moving window mechanism helps to reduce negative impact caused by malicious nodes. We attribute the results of the comparison to the fact that the beta reputation systems are not able to make distinction between the two classes of misbehaving nodes – selfish and malicious. In the Dirichlet reputation system, malicious nodes are always assigned higher cost than selfish nodes. This means that a very low probability of malicious behavior is always more costly than a very high probability of selfish behavior. In the beta reputation system, the cost of nodes depends on their relative probability of misbehaving, regardless of whether the nature of misbehavior is selfish or malicious. In case that a malicious node has a lower probability of misbehavior than a selfish node, the beta reputation system will assign a lower cost to the path containing that malicious node than to the one containing the selfish node. In the Dirichlet reputation system this could not happen, as there is a clear distinction between selfish and malicious behavior.

## 7. Conclusion

We propose a Dirichlet reputation system with moving window mechanism to address the problem of node misbehaviors in wireless *ad hoc* networks. Our solution makes distinction between selfish and malicious misbehavior, as opposed to reputation systems that are only able to classify nodes as good and misbehaving (bad). Our reputation system integrates the Dirichlet distribution with Bayes' theorem, which allows for finer granularity of node classification and updating of the reputation based on new knowledge. Our system is distributed and collaborative in a sense that nodes acquire new knowledge through first-hand

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec

observations of their neighbors' behavior, and through the reports they receive from other nodes containing their first-hand observations. We show that our system is able to improve network good throughput and decrease the number of maliciously modified or injected packets in the network when compared to the beta reputation systems.

## Acknowledgements

This work is partially supported by Tennessee Higher Education Commission's Center of Excellence in Applied Computational Science and Engineering.

## References

1. Buchegger S, Mundinger J, Le Boudec J. Reputation systems for self-organized networks: lessons learned. *IEEE Technology and Society Magazine, Special Issue on Limits of Cooperation in Wireless Communications* 2008; **27**(1): 41–47.
2. Rackley S. Wireless Networking Technology: From Principles to Successful Implementation, Newnes: 2007. ISBN-10: 0750667885.
3. Jøsang A. (ed). Trust and Reputation Systems. Foundations of Security Analysis and Design IV, Springer: Bertinoro, Italy, 2007.
4. Yang L, Kizza JM, Cemerlic A, Liu F. Fine-grained reputation-based routing in wireless *ad hoc* networks. IEEE International Conference on Intelligence and Security Informatics, 2007; New Brunswick, NY.
5. Zhang Y, Fang Y. A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks. *IEEE Transactions on Parallel and Distributed Systems* 2007; **18**(8): 1134–1145.
6. Marti S, Giuli TJ, Lai K, Baker M. Mitigating routing misbehavior in mobile *ad hoc* networks. MobiCom, ACM: Boston, MA, 2000.
7. Resnick P, Zeckhauser R. Trust among strangers in Internet transactions: empirical analysis of eBay's reputation system. The Economics of the Internet and E-Commerce. Michael R. Baye editor Volume 11 of Advances in Applied Microeconomics, 2002; Elsevier Science: Amsterdam, 127–157.
8. Resnick P, Zeckhauser R, Friedman E, Kuwabara K. Reputation systems. *Communications of the ACM* 2000; **43**(12): 45–48.
9. Cox LP, Noble BD. Samsara: honor among thieves in peer-to-peer storage. Proceedings of the 19th ACM SOSP. Bolton Landing, NY, 2003.
10. Cooper BF, Garcia-Molina H. Peer-to-peer data preservation through storage auctions. *IEEE Transactions on Parallel Distributed System* 2005; **16**(3): 246–257.
11. Buchegger S, Boudec JL. Performance analysis of the CONFIDANT protocol: Cooperation of nodes – fairness in dynamic *ad hoc* networks. IEEE/ACM Symposium on Mobile *Ad Hoc* Networking and Computing (MobiHOC), Lausanne: Switzerland, 2002.
12. Michiardi P, Molva R. CORE: a collaborative reputation mechanism to enforce node cooperation in mobile *ad hoc* networks. The IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security. 2002.
13. Xiong L, Liu L. A reputation-based trust model for peer-to-peer e-Commerce communities. *IEEE International Conference on E-Commerce Technology* 2003; 275.
14. Buchegger S, Boudec JYL. The effect of rumor spreading in reputation systems for mobile *ad hoc* networks. *In Proceedings of WiOpt Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* 2003.
15. Ganeriwal S, Balzano LK, Srivastava MB. Reputation-based framework for high integrity sensor networks. *ACM Transaction on Sensor Network* 2008; **4**(3): Article 15.
16. Wang Y, Giruka VC, Singhal M. A fair distributed solution for selfish nodes problem in wireless *ad hoc* networks. *Ad Hoc, Mobile, and Wireless Networks* 2004; Springer Berlin/Heidelberg. **3158**: 211–224.
17. Buttyan L, Hubaux J. Nuglets: a virtual currency to stimulate cooperation in self-organized *ad hoc* networks. Lausanne, Switzerland, Institute for Computer Communications and Applications, Swiss Federal Institute of Technology. 2001.
18. Buttyan L, Hubaux J. Stimulating cooperation in self-organizing mobile *ad hoc* networks. *Mobile Networks and Applications* 2002; **8**(5).
19. Rebahi Y, Mujica VE, Simons C, Sisalem D. SAFE: securing packet forwarding in *ad hoc* networks. 5th Workshop on Applications and Services in Wireless Networks. Paris, France. 2005.
20. Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian Data Analysis, Chapman & Hall/CRC: Boca Raton, 1995.
21. Buchegger S, Boudec JL. A robust reputation system for P2P and mobile *ad hoc* networks. In Proceedings of the Second Workshop on Economics of Peer-to-Peer Systems (P2PEcon), Harvard University: Cambridge MA, USA, 2004; June 2004.
22. Marti S, Garcia-Molina H. Taxonomy of trust: categorizing P2P reputation systems. *Computer Networks* 2006; **50**(4).

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/sec