Christian Gottron, André König, Ralf Steinmetz:
A Cross-layer Approach Towards Robustness of Mobile Peer-to-Peer Networks. In: Proceedings of the 7th IEEE International Workshop on Wireless and Sensor Networks Security (IEEE WSNS 2011), IEEE, Oktober 2011.

A Cross-layer Approach Towards Robustness of Mobile Peer-to-Peer Networks

Christian Gottron, André König, Ralf Steinmetz
Multimedia Communications Lab, TU Darmstadt, Darmstadt, Germany
{christian.gottron, andre.koenig, ralf.steinmetz}@kom.tu-darmstadt.de

***Abstract*—The lookup mechanism used to locate services in Peer-to-Peer systems can be attacked with little effort due to its decentralized and self-organizing nature. Security mechanisms aiming at rendering the lookup mechanism more robust mostly require a high amount of network resources. These mechanisms cannot be applied without adaptations when network resources are limited. In this paper, we introduce a novel approach to increase lookup robustness in mobile Peer-to-Peer networks. Here, network resources are limited by the mobile ad hoc network that is used as communication substrate. Our approach harnesses cross-layer information provided from the mobile ad hoc underlay to the Peer-to-Peer overlay. We derive analytical models to compare our approach to existing security mechanisms and validate our results by means of simulation. Our core findings show that our approach consumes less resources than existing mechanisms while the robustness remains at a comparable level.**

I. Introduction

Both Mobile Ad hoc Networks (MANET) and Peer-to-Peer (P2P) networks operate in a self-organizing way, without requiring central coordinating instances. Combining both architectures results in a completely decentralized system. Those Mobile Peer-to-Peer (MP2P) networks can be deployed in application scenarios without a reliable infrastructure as, e.g., in disaster relief operations. Several MP2P architectures have been proposed in recent years such as, e.g., *Mad Pastry* [1], *VRR* [2], or a mobile version of Chord [3]. These combine structured P2P systems based on Distributed Hash Tables (DHT) with a MANET underlay.

Providing a robust lookup mechanism for MP2P networks is challenging due to their decentralized and infrastructure-less nature. Security issues in MP2P networks had been largely neglected in the literature until now, as multiple security mechanisms for MANETs and P2P networks already exist. However, those security mechanisms may not be applicable, as MP2P networks introduce new challenges in particular to the P2P overlay. Contrary to the Internet underlay, resources in a MANET underlay are strongly limited. Due to this, new mechanisms are required to increase the robustness of the MP2P overlay without consuming a large amount of resources. Therefore, we propose a cross-layer based security mechanism, which provides a robust lookup mechanism on the overlay by harnessing underlay information. We evaluate our approach using the example of the *Incorrect Lookup Routing* attack [4].

The rest of the paper is structured as follows. We survey related work in the next section. In Section 3, we introduce our cross-layer approach. In Section 4, our approach is compared to traditional P2P security mechanisms analytically. In Section 5, we validate our analytical approach by means of simulation. In the last section, we provide conclusions and discuss our future work.

II. Related Work

In this section, we present attacks and security mechanisms that are closely related to our research and have motivated our work. As only few security mechanisms were developed specifically for MP2P networks, we also refer to algorithms proposed for MANETs and P2P networks, separately.

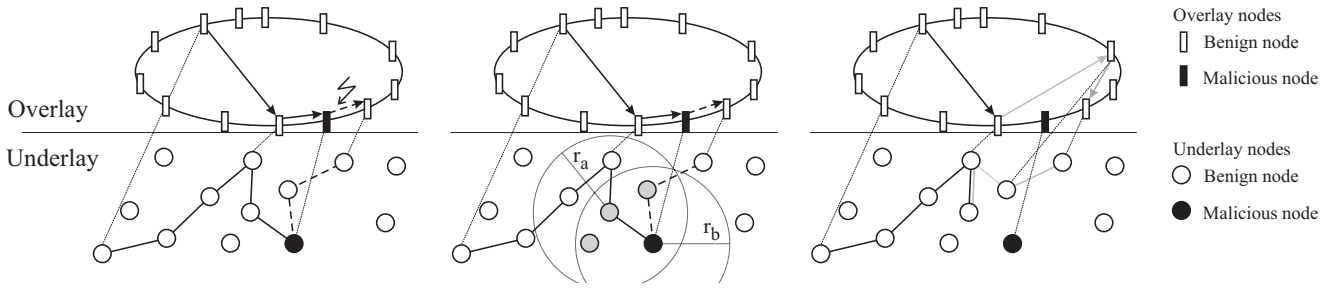
Sit et al. [4] introduced various attacks on DHTs, including the *Incorrect Lookup Routing* attack. During this attack, malicious nodes drop received lookup messages instead of forwarding them. In order to reduce the impact of this routing attack, they proposed an *Iterative Routing* algorithm. This algorithm provides routing information to the source, which is, therefore, able to respond to malicious behavior of intermediate peers.

Castro et al. [5] analyzed routing attacks on DHTs more extensively. They derived an equation to estimate the packet loss caused by the *Incorrect Lookup Routing* attack and introduced several security mechanisms as *Redundant Routing*. *Redundant Routing* is based on sending multiple lookup requests on parallel paths in order to increase the probability of a request reaching the destination intended.

Further approaches as *Resistant Routing* [6] or a combined redundant, iterative approach [7] were introduced to increase the robustness of the DHT routing algorithm. Though, those are mostly based on either *Iterative* or *Redundant Routing* (or on both of them) and all of those mechanisms increase the robustness at the cost of an increased routing overhead.

Besides increasing the robustness of the routing algorithms, using replicas is a further approach to improve the robustness of the lookup mechanism. Several distribution schemes for replicas were proposed such as a distribution of the replicas in the neighborhood of the responsible peer [5] or a uniform distribution of the replicas subject to the peer IDs [7].

For MANETs, multiple security mechanisms were introduced in the recent years. In order to prevent malicious behavior, cryptographic approaches such as ARAN [8], SEAD [9], SRP [10], and SAODV [11] were proposed. As those intrusion prevention systems may fail, intrusion detection systems, such as Watchdog [12] are able to detect malicious behavior. Watchdog observes messages received and sent by



(a) Malicious overlay node drops lookup request (b) Underlay neighbor nodes detect dropped lookup request (c) Underlay neighbor node informs previous intermediate overlay node

Fig. 1. Behavior of the cross-layer security mechanism during an *Incorrect Lookup Routing* attack

nodes in transmission range in order to identify malicious nodes. Further, trust-based approaches such as Pathrater [12] or the mechanisms proposed by Yi et al. [13] or Bahalaji et al. [14] and others, are able to exclude malicious nodes.

Though several MP2P architectures were introduced in recent years (e.g. [1], [2]), issues regarding network security and robustness were mostly neglected. Therefore, only very few mechanisms have been proposed to date. E.g., Kutzner et al. [15] briefly discussed security requirements for a specific MP2P algorithm.

III. Cross-Layer Approach for Securing MP2P Networks

MP2P networks introduce new security challenges such as strongly limited resources and a completely decentralized architecture. Due to this, traditional security mechanisms may not be applicable. In the following we introduce our approach which increases robustness against routing attacks by harnessing cross-layer information. We define our requirements and assumptions regarding the MP2P network. Thereafter, we discuss how our approach detects and responds to malicious behavior during an overlay lookup.

A. Assumptions and Requirements

Our scenario is based on a combined architecture of a DHT overlay with a MANET underlay. As discussed in Section 2, several mechanisms to increase the robustness and reliability of MANETs were proposed in the recent years. We assume that these are sufficient to provide robustness of the network underlay in MP2P scenarios since an MP2P architecture does not introduce any significant additional limitations to the underlay. Yet, most P2P security mechanisms against routing attacks are based on increasing the network's robustness by introducing redundancy. As the MANET underlay introduces strong limitations regarding the available bandwidth, those countermeasures are inefficient in MP2P networks. Due to this, we focus on increasing the robustness of the overlay in this paper.

We assume that each node in the MP2P network is assigned a unique, randomly distributed identifier, which is used for the overlay routing. Furthermore, multiple overlay hops are required in order to request an object in the DHT. With each hop, the request is forwarded to a node which is closer to the destination regarding the numerical distance between the unique identifier of the node and the requested object. This assumption is satisfied by most of the existing DHT and MP2P architectures. Further, each node in the MANET must also participate in the P2P network. This assumption is satisfied in disaster relief scenarios, where each node is preinstalled with the same software.

Furthermore, each node must be able to operate in promiscuous mode in order to overhear messages sent by neighbor nodes. The underlay links to neighbor nodes and the quality of those links has to be monitored in order to distinguish a broken connection from malicious behavior. Due to this, each node is aware of all nodes in transmission range and is able to detect when a node leaves the transmission range. Several MANET routing algorithms are able to provide this information. Protocols as BATMAN [16] or ETX-based algorithms [17] monitor the link quality in terms of throughput and packet loss. They derive routes not only based on the length regarding the number of hops between source and destination, but also based on the reliability of the route. However, most of the other routing algorithms can be easily adapted to provide this information.

The MANET underlay must be able to identify and observe overlay lookup messages. Underlay messages must provide information on the IP-address of the next underlay hop, the overlay identifier of the next overlay hop, a sequence number and the IP-address of the previous overlay hop. The IP-address of the next underlay hop should be provided by the MANET routing table. Further, the identifier of the next overlay hop can be extracted from the overlay lookup message. The underlay message can be easily extended in order to provide the sequence number and the IP-address of the previous peer.

B. Detection of Malicious Behavior

One of our major challenges is to detect malicious nodes in the MP2P network. During a lookup, the overlay request message must be forwarded by intermediate overlay nodes to reach the destination. Each of those overlay hops corresponds to a complete underlay route where the next overlay node is the destination. After the message reaches this overlay node, the route is completed for the underlay and a new route needs

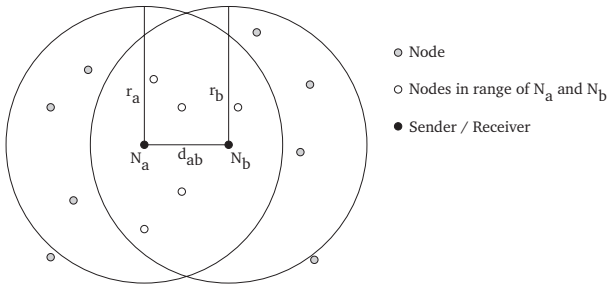


Fig. 2. An example for an underlay hop

to be established whenever this overlay node is not the final destination of the lookup. The Watchdog IDS [12] is a very promising approach, as no further bandwidth is required and, therefore, no overhead is introduced during the detection of malicious nodes. Yet, this security mechanism is only able to detect malicious behavior within an underlay route. It is not able to detect whether or not a message needs to be forwarded further after reaching the underlay destination. We need to adapt this security mechanism by harnessing cross-layer information to extend the benefit of the Watchdog IDS to the overlay.

In order to detect malicious behavior, each node has to monitor messages sent by the nodes in transmission range. By analyzing those messages, we are able to decide whether or not a message has been forwarded correctly. Whenever a lookup message is received by a peer, but is not forwarded, the underlay nodes are able to detect this malicious behavior as shown in Figure 1(b). Even though the underlay destination is reached, those underlay nodes may be used to detect malicious behavior in the overlay. Each node in transmission range of the malicious peer and the previous underlay node is able to detect this behavior. For this, those nodes have to compare the overlay identifier of the malicious node with the overlay identifier of the source and the destination of the lookup request. Whenever a node does not respond to a request and is neither the destination nor the source of this request, potential malicious behavior is detected. In order to avoid false positives, the link quality to nodes in transmission range has to be monitored to differ between malicious behavior and a lost link due to, e.g., node mobility.

Rerouted requests by malicious peers can also be detected by nodes in transmission range. Most DHT routing algorithms strictly define how to select the next hop node regarding the next hop's identifier. By checking whether those rules have been complied with, we are able to detect rerouted messages by comparing the identifier of the peer with the next hop node. An example is shown in Figure 2. The node N_a forwards a message to node N_b. Besides those two nodes, all other nodes that are within the transmission range of the sender can detect this transmission. Further, nodes which are in transmission range of the receiver of this message are able to detect whether this received message is forwarded by N_b. Therefore, all nodes which are in transmission range of both nodes, N_a and N_b, are able to detect a dropped message.

C. Reaction to Malicious Behavior

After detecting a malicious peer, our proposed security mechanism must counteract its behavior. Initially, a node must be chosen to respond to the attack. Thereafter, we have to define how this node responds to the attack.

As only nodes which are in transmission range of the intermediate underlay node which forwards the request and the malicious peer that drops the message are able to detect the misbehavior, one of those nodes must respond to the attack. In order to decide which of those nodes should respond, we have to consider whether or not the malicious nodes collude. In a scenario without colluding malicious nodes, the intermediate underlay node that forwards the message to the malicious peer should respond. This intermediate underlay node is able to detect the malicious behavior in any case as we assume bidirectional links. However, when the malicious nodes collude, we need to consider that this node also behaves maliciously. Therefore, each node that is in transmission range of the malicious node and that has received the request has to be considered as a responding node. This requires a coordination of those nodes as, otherwise, multiple nodes will respond to the attack and introduce an unnecessary traffic overhead. We propose that the node which has an identifier closest to the destination of the request should react to the attack. We use a waiting time that depends on the difference between the identifiers in order to assure that a node close to the destination forwards the message first. Each of the other nodes in the neighborhood are, therefore, able to detect that the request has already been forwarded by another node. This way, multiple retransmissions can be avoided.

Having detected misbehavior ongoing and having determined which node should respond to the misbehavior, we differentiate between two response mechanisms. The first mechanism is based on the assumption that each node in the MANET is also participating in the P2P network. In this case, a node which has detected malicious behavior is able to directly initiate a new lookup request. This way, we reduce the cost for the new request in terms of hops required because the request is re-initiated by the intermediate node that is closest to the destination and not by the originator. Further, we avoid loops by using a unique sequence number. This approach is simple to implement.

In our second approach, the node that detects the attack sends a notification message to the previous peer. This node then selects another next hop for forwarding the request as shown in Figure 1(c). This way, the average number of hops per request is only increased by a single overlay hop whenever a request is forwarded to a malicious node.

D. Open Challenges

Some challenges regarding the cross-layer security mechanisms for MP2P networks remain unsolved. A node must decide whether or not the responding node is the intended destination of the request or if the node only claims to be the destination and tries to deny the request. Castro et al. [5] introduced a mechanism which compares the ID density of

the virtual neighborhood of the sender of the request with the distance between the ID of the responding node and the ID of the requested object/node. Whenever this density and the distance differ significantly, a malicious behavior of the responding node is assumed. This mechanism may also be used in the context of MP2P networks. Yet, this would introduce a large overhead as each node has to determine the ID density in the neighborhood. Furthermore, each node in the network has to be able to validate the mapping of overlay identifiers to IP-addresses. Otherwise, a malicious node may misuse this by forging overlay identities and map them to any existing or non-existing IP-address during forwarding.

To tackle these challenges, in our future work, we plan to measure the overhead generated by the mechanism introduced by Castro et al. . By harnessing information provided by the MANET routing algorithm we may be able to measure the node density without any overhead at all. Further, signatures are required that guarantee the correct mapping of node identities.

IV. Analytical Evaluation

We evaluate the performance of our our cross-layer approach for counteracting the *Incorrect Lookup Routing* attack in this section using an analytical approach. As metrics, we use (1) the packet loss, (2) the number of sent messages per lookup process, (3) the average required number of overlay hops per lookup process, and (4) the delay between initiating a lookup request and receiving the answer. The packet loss has the highest impact on the robustness of the lookup mechanism and is, therefore, the primary metric. As the bandwidth is strongly limited, the number of sent messages per lookup has to be further considered to estimate the efficiency of the algorithm. Further, the number of sent messages is a function of the average required number of overlay hops per request. Our last metric, the delay, has to be considered in specific application areas with real-time constraints. In this section, we compare the characteristics of our approach with the basic *traditional* P2P security mechanisms.

A. Recursive Routing

Most DHT routing algorithms are based on a recursive approach. Requests are, in each step, forwarded to a node closer to the destination regarding the virtual distance in the ID space until the destination is reached. However, this approach does not result in a robust lookup mechanism. Castro et al. derived an equation to estimate the probability of a successful lookup process σ_{Rec}. This equation is based on the required number of overlay hops h and the fraction of malicious nodes f as shown in Equation 1. The number of overlay hops required is defined by the particular P2P routing algorithm and, for DHTs, is a logarithmic function of the network size.

$$\sigma_{Rec} = (1-f)^h \tag{1}$$

Besides the fraction of lost requests, the number of sent messages m_{Rec} per request and, therefore, the number of required hops must be considered. Those two metrics correlate and can also be used to estimate the introduced traffic per request. Further, we have to differentiate between the number of average hops of a successful request h and the number of average hops h_{Rec} that include also failed requests. In a recursive scenario without any security mechanisms, the average number of sent lookup messages m_{Rec} is equal to the number of hops that includes the failed requests h_{Rec}. The delay is a direct function of the average number of hops h. The average number of hops and sent messages can be calculated as shown in Equation 2.

$$m_{Rec} = h_{Rec} = \sum_{i=0}^{h-1} (1-f)^i \tag{2}$$

B. Traditional P2P Security Mechanisms

As discussed in the related work section, several security mechanisms were introduced to increase the robustness of MANETs and P2P networks. We focus on the *Redundant Routing* and the *Iterative Routing* algorithms, as those are two of the basic security mechanisms for P2P networks and most of the more recent mechanisms are based on those algorithms.

Redundant Routing is based on the recursive algorithm but increases the robustness by sending s route request messages in parallel. Therefore, the probability of a successful lookup σ_{Red} is increased compared to σ_{Rec} as a request only fails whenever all s requests are routed to a malicious node. Yet, as long as no replicas are distributed in the network, the request fails whenever the destination behaves maliciously. The probability of a successful lookup can be described by Equation 3.

$$\sigma_{Red} = (1-f) * (1-(1-(1-f)^{h-1})^s) \tag{3}$$

The number of average hops is still described by Equation 2. Yet, the number of sent messages differs, as s messages are sent in parallel. Therefore, m_{Red} is defined by the average number of hops h_{Rec} including the failed requests and the number of parallel requests s as shown in Equation 4. Further, the delay is a direct function of the average number of hops h, similar to the recursive algorithm, as the requests are sent in parallel.

$$m_{Red} = h_{Rec} * s \tag{4}$$

During an *Iterative Routing*, intermediate nodes send reply messages to the source of the request instead of forwarding the request. Those reply messages provide a set of next hop nodes to the source node of the request. Therefore, the source is able to control the routing of the request directly and is, therefore, able to respond whenever a request is dropped or misrouted. However, as each intermediate peer provides only r next hop addresses, we assume that the request is finally dropped when all r nodes behave maliciously. Although this improves the fraction of successful lookup process σ_{Iter} strongly, as described in Equation 5, at least the destination of the request has to behave benignly.

$$\sigma_{Iter} = (1-f) * (1-f^r)^{h-1} \tag{5}$$

The required number of hops increases due to the iterative behavior of the routing algorithm as described in Equation 6. Due to this, also the number of required messages per request increases. Further, due to the reply messages sent by each hop, the delay increases and is at least twice as high as for the recursive routing algorithm. The delay increases furthermore whenever a request is dropped.

$$m_{Iter} = h_{Iter} = \sum_{i=0}^{h-2}(\sum_{j=0}^{r} f^j(1-f))^i * (\sum_{k=0}^{r} f^k + f^k * (1-f)) \tag{6}$$

C. Cross-Layer Approach

Our proposed security mechanism harnesses cross-layer information in order to increase network robustness. To reduce overhead, we introduce redundancy only when required, whenever a lookup request is dropped by a malicious node. As our mechanism is based on observing neighbor nodes, we require nodes in transmission range of malicious nodes to detect and respond to malicious behavior. We consider the example shown in Figure 1(b), where the black peer receives and drops a request. In this particular situation, the three gray underlay nodes are able to detect this malicious behavior. The request can be sent to another intermediate peer unless all three nodes behave maliciously. Due to this, the parameters do not depend only on the fraction of malicious nodes and the number of required hops, but also on the field size, the overall number of nodes, the transmission range, and the distance between sender and receiver as well as on the relation of the distance and transmission range to the number of available intermediate nodes N_X.

Because of limited space, we focus on our second approach in which the node that detects the attack sends a notification message to the previous peer. Due to this mechanism, we do not have to repeat the lookup process from the very beginning each time a malicious behavior is detected, but only have to repeat the last step of the lookup. Therefore, we assume this approach as more promising.

We define n as the number of nodes which are in transmission range of both nodes, the underlay sender and receiver (i.e., of Nodes N_a and N_b in Figure fig:cm). The probability of a successful lookup is similar to an iterative approach, as long as a sufficient number of neighbor nodes n is available as described in Equation 7.

$$\sigma_{CL} = (1-f) * (\sum_{i=1}^{r}(\sum_{j=1}^{n} f^j * (1-f))^i * (1-f) + (1-f))^{h-1} \tag{7}$$

The number of average messages per lookup is described by Equation 8. The number of average hops h_{CL} is equal to the number of average sent messages m_{CL} and is a function of the fraction of malicious nodes f, the number of neighbors n, and the number of stored addresses per routing table entry r.

$$m_{CL} = h_{CL} = \sum_{i=0}^{h-2}(\sum_{k=1}^{r}(\sum_{l=1}^{n} f^l)^k * (1-f)^k + 1 - f)^i * (\sum_{k=1}^{r}(\sum_{j=1}^{n} f^j * (1-f))^k * (1-f) + (1-f))^{h-1} \tag{8}$$

Regarding the delay, we achieve values comparable to a redundant or recursive approach. Yet, the delay may be increased whenever packets are dropped during the lookup.

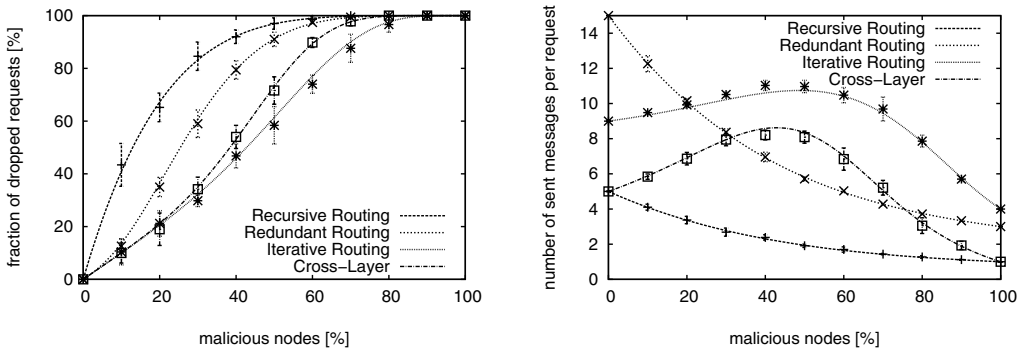
V. Evaluation

In order to validate the analytical models, we implemented a tool that simulates the lookup process in an abstract way. For this, the node behavior and the routing algorithm including the different security mechanisms were implemented. Yet, we neglected the effects of the underlay including the node mobility. The tool simulates 100 lookup processes and calculates the average number of hops and the probability of a successful request. The number of overlay hops between sender and receiver h, and the parameters for the security mechanisms have to be given for the simulations. Both the results of the simulator and results of the equations are comparable. From this, we conclude the accuracy of the analytical description presented. The simplified algorithm for the simulation of the *Redundant Routing* is shown in Algorithm 1 as pseudocode.

We evaluated all four routing mechanisms by an analytical approach and by simulation. Due to space limitations, we present selected results for a scenario with an average overlay hop count of $h = 5$ that is representative for all results obtained. The number of requests sent in parallel (s), the number of stored addresses per routing table entry (r) and

Algorithm 1 Pseudocode for the simulation of a redundant lookup

```
successful_transmission = false
for number_of_parallel_lookups do
  while destination_not_reached do
    if random_value < fraction_of_malicious_nodes
    then
      message dropped
      break
    else
      if node = destination then
        overall_hops ← overall_hops + 1
        successful_transmission = true
      else
        overall_hops ← overall_hops + 1
        forward_message
      end if
    end if
  end while
end for
```



(a) Efficiency in terms of dropped requests

(b) Overhead in terms of messages sent

Fig. 3. Comparison of the performance of the cross-layer security mechanism and traditional mechanisms

the average number of neighbors are set to 3. All of the three robust mechanisms were able to decrease the packet loss, as shown in Figure 3(a). Those plots show curves predicted by the analytical models and 99 percent confidence intervals obtained from the simulation studies. Both the iterative and our cross-layer approach respond actively to malicious behavior whereas a recursive or redundant algorithm provides no robustness or only a robustness by redundancy. Therefore, the iterative and the cross-layer approach are more efficient regarding the number of dropped requests. However, the number of sent messages of our cross-layer approach is low compared to the iterative approach as shown in Figure 3(b). In scenarios without any or with a small fraction of malicious nodes, the number of sent messages is comparable to the traffic introduced by a *Recursive Routing*.

VI. Conclusions and Future Work

We presented a novel approach to detect and respond to malicious behavior in an MP2P scenario. This approach is adapted to the challenges introduced by an MP2P scenario such as limited resources and a completely decentralized network. In the analytical models described above, effects as the mobility or the heterogeneity of a realistic network regarding the transmission range of different nodes have been neglected by now. The node mobility and, therefore, the dynamic behavior of the topology will have an impact on the packet loss. Therefore, the fraction of transmission loss generated due to mobility has to be added to the fraction of malicious nodes. This can also have an impact on our approach when e.g. an intermediate node detects a transmitted message but leaves the transmission range of the receiver of this message at exactly this time. As a result, a dropped message is detected even when the receiver of this message behaves benignly. However, by monitoring the link quality, we should be able to detect a broken link. Therefore, we should be able to reduce the number of false positives generated by our cross layer approach to a minimum. Also the impact of the heterogeneity of the transmission range should be quite low, as many MANET routing protocols use only bi-directional routes by design. Further, our approach is capable of harnessing intermediate nodes which are in transmission range of both sender and receiver to detect malicious behavior. Due to this, we should be able to detect malicious nodes even when uni-directional routes were used.

In our future work, we will address the unsolved challenges stated above. Furthermore, a more intensive simulation study, an implementation of our approach, and a testbed evaluation will be our next steps in order to reveal effects which may be unnoticed by now. Especially effects of mobility and an optimization regarding the energy consumption will be considered in our future research.

References

[1] T. Zahn and J. Schiller, "Madpastry: A dht substrate for practicably sized manets," in *Proc. of the 5th IEEE ASWN*, 2005.

[2] M. Caesar et al., "Virtual ring routing: network routing inspired by dhts," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, 2006.

[3] C. Cramer and T. Fuhrmann, "Performance evaluation of chord in mobile ad hoc networks," in *Proc. of the 1st MobiShare*, 2006.

[4] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Proc. of the 1st IPTPS '01*, 2002.

[5] M. Castro et al., "Secure routing for structured peer-to-peer overlay networks," in *Proc. of the 5th OSDI*, 2002.

[6] M. Sànchez-Artigas et al., "Making replication secure over structured p2p systems: Defending against omission attacks," in *Proc. of the 5th DBISP2P*, 2007.

[7] K. Hildrum and J. Kubiatowicz, "Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks," in *Proc. of the 17th DISC*, 2003.

[8] K. Sanzgiri et al., "A secure routing protocol for ad hoc networks," in *Proc. of the 10th IEEE ICNP*, 2002.

[9] Y.-C. Hu et al., "Sead: secure efficient distance vector routing for mobile wireless ad hoc networks," in *Proc. of the 4th IEEE WMCSA*, 2002.

[10] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. of the CNDS*, 2002.

[11] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. of the 1st ACM WiSE*, 2002.

[12] S. Marti et al., "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of the 6th MobiCom*, 2000.

[13] S. Yi et al., "Security-aware ad hoc routing for wireless networks," in *Proc. of the 2nd ACM MobiHoc*, 2001.

[14] N. Bhalaji and A. Shanmugam, "Reliable routing against selective packet drop attack in dsr based manet," *Journal of Software*, vol. 4, no. 6, 2009.

[15] K. Kutzner et al., "Securing the scalable source routing protocol," in *Proc. of the WTC 2006*, 2006.

[16] M. Ikeda et al., "A bat in the lab: Experimental results of new link state routing protocol," in *Proc. of the 22nd AINA*, 2008.

[17] D. S. J. De Couto et al., "A high-throughput path metric for multi-hop wireless routing," *Wirel. Netw.*, vol. 11, July 2005.