

Experiences from the MANIAC Challenge

Heiko Will Felix Juraschek Mesut Güneş Jochen Schiller
Distributed Embedded Systems
Institute of Computer Science
Freie Universität Berlin and Humboldt University Berlin, Germany
{heiko.will, felix.juraschek, mesut.guenes, jochen.schiller}@fu-berlin.de

Abstract—The MANIAC Challenge is a competition for cooperation strategies in wireless ad-hoc networks with the focus on experimental evaluation. We present the results of the MANIAC Challenge and discuss characteristics of real networks such as link instability and mobility, which are often simplified in common network simulators. We introduce our strategy *Friendly Clustering* that won the *Performance Award*. *Friendly Clustering* is based on monitoring the neighbors' forwarding behavior and assessing their willingness to relay future data packets. Further on, we describe the challenges in implementing the strategy for real networks, such as the DES-Testbed at the *Freie Universität Berlin* and the ad-hoc network of the MANIAC Challenge.

Index Terms—wireless mesh network, cooperation strategies, practical testbed experience, multi-hop routing

I. INTRODUCTION

Mobile ad hoc networks (MANETs) have many characteristics, such as limited energy and a dynamic topology due to a high degree of mobility. The multi-hop topology requires the participating nodes to forward traffic destined to other nodes without a direct benefit for the forwarding node. Considering the energy usage of transmissions, forwarding is an expensive task. Therefore, a selfish behavior is encouraged that may result in dropping packets. However, without any cooperation in routing, network communication would stop and the network would cease to exist. As a result, each node relies on the cooperative behavior of other nodes to use network services.

Optimized Link State Routing (OLSR) is a pro-active routing protocol [1] that is often used in MANET deployments, such as urban community networks like Freifunk [2] to provide Internet access. The proactive OLSR establishes routes to all reachable nodes before any data packets. In the basic configuration of OLSR, nodes forward all packets according to these routes, selfish behavior of individual nodes is not supported or desired.

The *Mobile Ad Hoc Networking Interoperability and Cooperation* (MANIAC) Challenge, designed by the *Virginia Tech* and *Bucknell Universities*, is an experimental research approach in form of a competition of different cooperation strategies for routing in MANETs [3] [4]. The participating teams design a routing strategy and implement it for a real network using off-the-shelf notebooks with IEEE 802.11 network devices. The provided MANIAC API allows to handle incoming packets. They can either be forwarded, dropped, or sent to a different node than suggested by the kernel routing table.

The MANIAC Challenge is part of a trend in the wireless research community, in which testbeds as an experimentation environment gain more significance. Up to today, most of the research on wireless networks is based on simulations [5]. Simulations prove useful to develop and analyze network protocols, especially in regard to scalability. However, results obtained from simulations can not be transferred directly to real world network deployments, since the degree of realism of current simulators is restricted by simplified models [6].

For this reason, many wireless testbeds based on different networking technologies have been set up in the last decade. The testbeds can be classified into *persistent* and *non-persistent* ones. Persistent testbeds consist of a stationary backbone and are deployed once. Mobile nodes are added on demand if required by a particular experiment. The *Distributed Embedded Systems* (DES)-Testbed is such a testbed. It enables research on manifold network architectures but scenarios with mobile nodes require additional efforts.

In contrast, non-persistent testbeds are deployed on demand for each experiment. Therefore, non-persistent testbeds are more suited for studies with node mobility. However, the repeatability of experiments is more difficult, because for each repetition of an experiment, the node movement has to be exactly repeated. The ad-hoc network which is set up by the participants of the MANIAC Challenge is such a non-persisting testbed. All nodes are allowed to move around freely during the competition.

The contribution of this paper is twofold. We describe characteristics of real networks such as link instability and mobility, as experienced in the MANIAC Challenge and in the DES-Testbed. Additionally, we present the *Friendly Clustering* strategy, which won the Performance Award of the MANIAC Challenge 2009.

The remainder of this paper is structured as follows. In Section II we describe the MANIAC Challenge and the presented cooperation strategies. We introduce the *Friendly Clustering* strategy in Section III, and the challenges in implementing it using real hardware. In Section IV, we discuss the experience and results with *Friendly Clustering* on the DES-Testbed and in the MANIAC competition. We finish the paper with conclusions for the next MANIAC Challenge.

II. THE MANIAC CHALLENGE

The goal of the MANIAC Challenge is to show that a MANET can continue to work even if it contains a few selfish nodes. Selfish behavior is expressed by dropping traffic in order to save energy or to free bandwidth. Another goal is to capture mobility traces in form of changes in the routing topology during the competition. These traces are available for the research community on the project website for further analysis and as an input to simulation scenarios.

The first challenge with 5 teams took place as part of the *IEEE Globecom* in 2007. The second competition with 8 teams followed at the *IEEE PerCom* in Galveston, Texas, held in March 2009.

A. Competition setup

Each team adds two nodes consisting of a notebook equipped with a IEEE 802.11g *wireless network interface card* (WNIC) to the network. The teams are free to choose their laptops and network adapters. Most teams chose WNICs based on Atheros chip sets, since they support the *promiscuous* mode required for sniffing data traffic. Additionally, four *MANIAC Source Nodes* provided by the organizers join the network. These source nodes generate unicast UDP traffic flows to the participating nodes. A node earns 10 points for each successfully received packet of a flow destined to the node. 1 point is lost for each forwarded packet. Half of the packets are classified as real-time packets, which means that they had to be successfully received in a short time span in order to get points. Points are accounted in three competition runs, each lasting 20 minutes.

On all nodes the OLSR daemon of the Naval Research Lab `nrlolsrd` [7] runs for route discovery and maintenance. Teams are not allowed to manipulate or

inject fake OLSR *HELLO* messages into the network in order to change the routing topology. A live visualization of the OLSR routing topology is available throughout the competition using the *Monitor for Mobile Ad hoc Networks* (MMAN) [8].

Teams implement their cooperative routing strategies using the MANIAC API. The MANIAC API resides on top of the kernel routing table maintained by the OLSR daemon and allows to adapt the routing behavior of the nodes. For each incoming packet, a node can decide to drop it, to forward it as suggested by the routing table, or to change the next hop to a different destination. In addition, nodes may sniff traffic of their neighborhood using the promiscuous mode of the network adapter.

All nodes move around freely during the competition. The MANIAC Challenge 2009 took place in a three story conference building, which they were also able to leave.

B. Cooperation strategies

The strategies in 2009 can be classified into *ratio-dropping* and *monitoring* strategies. With the first one, a node drops a fixed percentage of packets randomly regardless of the source or destination of the packet and of the behavior of the sending or receiving node. These kinds of strategies are easily implemented but also easily discovered by other nodes. As a counter measure to these strategies, all packets from and to nodes applying these strategies can be dropped, because the nodes do not monitor their neighborhood and therefore are not aware of the dropping and do not punish at.

Strategies of the monitoring class try to estimate the cooperativeness of their neighbors by considering the received and sniffed packets. Usually the nodes use a local score function to estimate the neighbor's cooperativeness based on the gathered information. The forwarding decisions are then based on the score of the destination node. Teams pursuing the monitor strategy approach modified well-known problems of game theory for their strategy, such as *tit-for-tat* [9] and the iterated prisoner's dilemma (IPD) [10] problem.

A team of the *Virginia Tech University* implemented the *tit-for-tat* strategy for both nodes. The *University of Cairo* implemented the *Mongoose* strategy which is a mix between observing the neighborhood and random forwarding decisions [11]. The *Strategy Award 2009* for to the most interesting strategy regardless of the performance in the competition, was awarded to Marcello Caleffi, who implemented *A Diversity Adaptive Approach for Cooperative Behavior* [12]. In his strategy the two team nodes have different roles. One acts as a ratio-dropper by dropping all packets whereas the

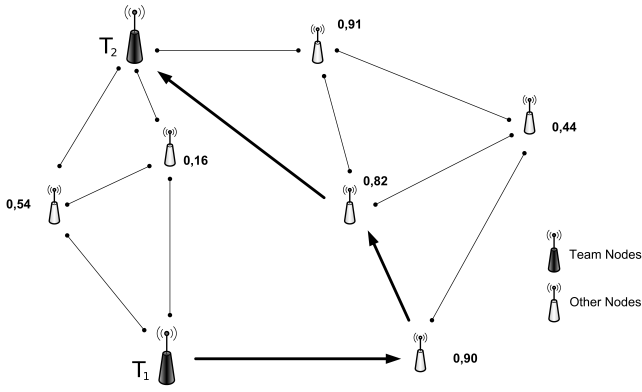


Figure 1: In this snapshot, T_1 sends a packet addressed to its team node T_2 over a 3-hop path. Although a 2-hop path exists, the 3-hop path is preferred, since the nodes on this path seem to be more cooperative in relaying packets.

other node applies the tit-for-tat strategy for detected cooperative nodes.

III. FRIENDLY CLUSTERING

The *Friendly Clustering* strategy is a monitoring strategy consisting of two parts: a *forwarding estimation scheme* and a *traffic classification* based on the packet destination. The forwarding estimation scheme derives a forwarding probability or score for each neighbor by monitoring all transmissions using the promiscuous mode of the WNIC. This score is used to divide the neighborhood into two disjoint sets: *cooperative nodes* N_c and *uncooperative nodes* N_u . Thus, the 1-hop neighborhood of a node t is the union of these two sets: $N(t) = N_c(t) \cup N_u(t)$. The partition is done by treating the best k nodes as cooperative and the remaining as uncooperative. This way, there are always neighboring nodes, which we consider cooperative and we do not have to calculate and balance out fixed thresholds.

Incoming and sniffed packets are rated with different weights based on their destination in the score calculation. This way we can apply a higher weight to packets addressed to our team node than to unknown nodes. To calculate the forwarding probability for a neighboring node n the following function with the weights u, v, w with $u + v + w = 1$, and the ratio of the particular packet counts C is used:

$$\text{score}(n) = u \cdot C_{tu}(n) + v \cdot C_{fu}(n) + w \cdot C_{to}(n)$$

$C_{tu}(n)$ stands for the packet ratio of forwarded packets to us, $C_{fu}(n)$ for the ratio of packets forwarded from us, and $C_{to}(n)$ for the ratio of packets forwarded to other nodes. Due to the different weights, the mechanism is

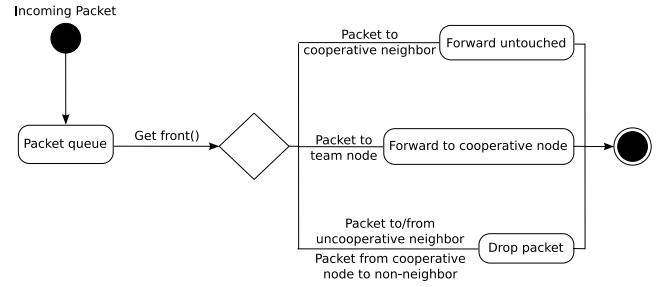


Figure 2: State diagram of handling packets from the *input queue*.

highly adaptive: a node dropping many packets destined to our team will quickly lose its cooperative status. Figure 1 shows a snapshot of a network with the node scores.

The forwarding decisions are based on the node score and the packet destination as depicted in Figure 2. Incoming packets addressed to nodes in the cooperative set N_c will be forwarded in expectancy of equal treatment. Packets destined to uncooperative nodes of N_u will be dropped in order to save energy. All other traffic will be dropped. Cooperative neighbors will sense our retransmission and tend to regard us as cooperative, if their strategy is also based on an estimation of cooperativeness. The strategy uses a hysteresis, keeping the score on a neutral value for a short time interval. This is useful, since traffic flows are not expected parallel in time and to initialize the trust relationship for unknown nodes.

We periodically exchange the gathered forwarding probabilities between our two team nodes. This way both nodes already have scores for nodes that enter the 1-hop neighborhood and previously have been neighbors of the other team node.

IV. PRACTICAL EXPERIENCE

In this section we first present the challenges in implementing Friendly Clustering in a real network environment. We present the validation of the strategy using simple scenarios in the stationary DES-Testbed. Finally, we discuss the results of the MANIAC Challenge and describe the network characteristics and their impact.

A. Implementing Friendly Clustering

For gathering detailed forwarding statistics of the neighbor nodes, we have to examine all received and sniffed packets. The MANIAC API makes the incoming packets accessible via the *input queue*, which has to be polled periodically. A second queue, the *sniff queue*, provides all sniffed packets which can be examined to

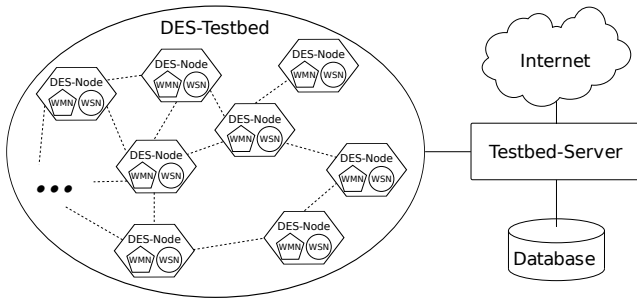


Figure 3: Architecture of the DES-Testbed. Each DES-Node consists of a wireless mesh router and a wireless sensor node. The stationary DES-Nodes are located in different buildings and are connected to the testbed server (DES-Portal) via Ethernet.

derive information of the current neighborhood. The queues are realized using the *libpcap* library.

During the implementation process, we discovered that the two queues are not synchronized. It may happen, that while a node handles a packet, further packets arrive in the input and sniff queues. Since only UDP traffic is sent, the order of arrival of the packets can not be determined. Therefore, the task of precise tracking of packets is difficult, especially when we try to observe if a particular neighbor forwards an already tracked packet.

We solved the problem by implementing a packet-based tracking module. This module processes every sniffed packet and determines if another transmission of this packet is to be expected. This is the case, when it has not reached its destination yet and the next hop is a neighbor node. If we do not overhear a following transmission from our neighbor, the packet likely has been dropped. This mechanism is especially useful to test neighbors if they forward packets received from us.

For the team node communication it is allowed to open a specific port and exchange messages. These messages are always forwarded and can not be dropped, since they do not enter the incoming queue. Therefore, the team communication is very reliable in contrast to the MANIAC traffic flows.

B. DES-Testbed

For the MANIAC Challenge preparation we tested a wide range of scenarios on the DES-Testbed. The DES-Testbed is a wireless mesh testbed with currently 110 nodes. The network architecture is depicted in Figure 3. A detailed description of the DES-Testbed architecture and the the research focus is available in [13], [14].

The goal of the evaluation of the DES-Testbed was to validate the monitoring and forwarding estimation mechanisms of *Friendly Clustering*. In contrast to the

network at the MANIAC Challenge, the DES-Testbed mesh routers are stationary. The stationary set-up allows to replay created scenarios easily, which is useful to evaluate the effect of specific settings for the weight parameters in the presented formula. We used the optimum weight parameters derived from the scenarios on the DES-Testbed for the competition in the MANIAC Challenge. In later scenarios we simulated basic mobility by switching nodes on and off.

For the scenarios in the DES-Testbed we selected up to 20 nodes and deployed *Friendly Clustering* on two of these nodes with a distance of at least two hops. We selected 4 source nodes at the network borders sending random UDP traffic flows to all participating nodes. On all other nodes we deployed ratio dropping strategies using drop percentages between 0-100%, which also varied over time to test if our approach is adaptive. On our team nodes, all received or sniffed packets were used to calculate node scores according to the presented function. As a result, our strategy was able to determine the nodes behavior, so that the calculated score for nodes with a low drop rate was significantly higher compared to nodes with a high drop rate. For the weight parameters $u = v = w$ delivered the best results, which was expected since the nodes dropped only a fixed percentage and do not take the origin or destination of a packet into account. Still, in the MANIAC Challenge other teams will likely consider their forwarding decisions on these values. Therefore we adjusted the values so that $u, v > w$, meaning that packets from or to us have a higher influence on the score.

The scenarios revealed an interesting weakness of our strategy resulting from the *hidden node* problem as pictured in Figure 4. It can happen, that we receive continuously packets from one of our neighbors and therefore regard it as cooperative. But it may be the case, that this node only forwards a small fraction of all packets to us. If we are not able to sniff the packets of the prior hop, there is no way of attaining the real amount of packets originally destined to us. Therefore, we also do not know the ratio of dropped and forwarded packets of this particular neighbor. Still, the monitoring of the node includes packets to our partner node and arbitrary nodes as well and it is likely that we will have more monitoring data concerning this node which will weaken this effect.

C. MANIAC Challenge

During the MANIAC Challenge all network nodes roamed the Galveston conference building. The organizers carried the source nodes into opposite directions

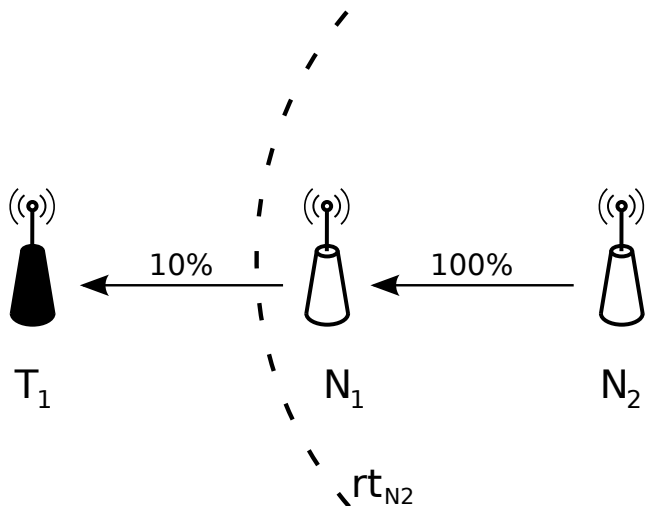


Figure 4: Effects of a hidden node for Friendly Clustering. The nodes N_2 and N_1 are on the path for a traffic flow t to our team node T_1 . N_2 forwards 100% of all packets to N_1 , whereas N_1 forwards only 10% of this packets to our team node T_1 . Still, T_1 will consider N_1 to be cooperative since T_1 is not inside the transmission radius rt_{N_2} of N_2 and therefore can not know if node N_1 dropped any packets.

Team	Strategy	Points
Freie Universität Berlin	Friendly Clustering	417,983
University of Cyprus	FiftySixKei	340,675
Virginia Tech	Tit-For-Tat	255,142
UNC Charlotte	Diverse Strategies	249,892
Uni. Napoli Federico II	Diversity Approach	244,673

Table I: Results of the MANIAC Challenge. For each received packet, a team earns 10 points, while for each forwarded packet it loses 1 point. The final points are accumulated over 3 competition runs each lasting 20 minutes.

to get far away of each other. The created MANET consisted of 20 nodes, with the longest observed route being seven hops long. The three competition runs were performed shortly after each other, leaving only short time to refine the strategies. The final results of the MANIAC Challenge 2009 are displayed in Table I. The table shows that the first five places are taken by strategies of the monitoring class. It was expected that monitoring strategies outperform simple ratio-dropping strategies since the latter are unable to adapt to their neighbor's behavior.

A closer look at our log files revealed the interesting fact that we received more than half of the packets directly from the source nodes. Therefore, these packets have not been forwarded by any competing nodes and

were received independent of the competitors strategies. This can be credited to the positioning of our nodes close to the source nodes. We developed a tool that displays live information of the network topology by showing all known nodes and their current scores. Additionally, nodes in our 1-hop neighborhood are highlighted. To be as much independent of other nodes as possible, we tried to maximize the number of source nodes inside our 1-hop neighborhood. The results show that node positions can have a huge effect on the received service in MANETs. This observation is also applicable to MANETs besides the MANIAC Challenge, in which node mobility is used to get a better position regarding the link quality, to decrease interference effects and the distance to gateway nodes.

However, for the MANIAC Challenge it is desired that the strategies have a big effect on the routing process and therefore single hop deliveries from source to destination node should be avoided. Therefore, we suggest for the next MANIAC Challenge to introduce a requirement for traffic flows such that source nodes send packets on a minimum route length M_r with $M_r > 1$. This way, it is ensured that each correctly received packet was delivered at least via two hops and has been forwarded by another competing node.

Another observation concerns the high amount of low quality links characterized by low received signal-strength indicator (RSSI) values that had a huge impact on the network performance. Over these links, it was possible to receive occasionally the broadcasted OLSR *HELLO* messages, which resulted that the senders were considered directly reachable by the OLSR daemon. However, due to the low link quality, these links were usually not usable for the unicast packets of the MANIAC traffic flows. There are two reasons for this behavior. First, whereas broadcast messages are usually sent with a data-rate of 1 Mbit/s in 802.11b/g, the unicast data-rates are usually higher. The auto-rate algorithm is supposed to detect this and decrease the used data-rate, but need several transmission to adapt the rate accordingly. Second, the unicast packets usually have a bigger packet size than the OLSR *HELLO* messages. Since the packet loss rate also depends on the packet size, it is much more likely that the unicast packets of the MANIAC traffic flows can not be correctly received even though an OLSR *HELLO* messages was received.

A similar effect arose from asynchronous links in the network. We received OLSR *HELLO* messages of certain nodes, but when we tried to send unicast data to these nodes, the ARP request was not answered and therefore, the OLSR daemon removed them from our routing table and the data flow ceased. This situation

repeated when our node received the next OLSR *HELLO* message. As a result of these phenomena, we had a very dynamic routing table as nodes entered and left our 1-hop neighborhood with a high frequency and we observed a high packet loss on these unstable links.

These unstable links had such a big impact on the network performance because the hop-count metric is used in the `nrlolsrd` implementation. Several studies on link metrics for wireless mesh networks have shown that the hop-count metric has several drawbacks [15]. As a result, different metrics such as the *Expected Transmission Count* (ETX) have been developed which take the forwarding probabilities of both directions into account by sending probe packets periodically. Other OLSR implementations allow the usage ETX, which favors routes with high quality links over shortest path routes. Since the usage of the ETX metric reduces the effect of unstable links and it would be an interesting modification for the next MANIAC Challenge.

Although the notion of unstable, low quality links and their effect on many routing algorithms is not new, many simulations still rely on a binary link model. This means that two nodes of the network can either hear themselves and receive all packets or not, in which case no packets are received. We conclude from the experience at the MANIAC Challenge that asynchronous links need high attention in wireless network research independent of the experiment environment.

V. CONCLUSION

We presented the results of the MANIAC Challenge 2009 and our strategy *Friendly Clustering* that won the *Performance Award*. The results of the competition indicate that monitoring approaches outperform simple ratio-dropping approaches. Although it is a labor-intensive task to implement a cooperation strategy for MANETs, interesting characteristics have been observed that are usually not sufficiently supported in simulators. This includes the link instability, asynchronous links with their influence on the routing topology, and the importance of node positioning in regard to special nodes like gateways or service providers. Improvements for the next MANIAC Challenge, such as the minimum route requirement and the usage of the ETX routing metric for OLSR, have been discussed. The MANIAC Challenge experiment delivered many insights in MANETs and hands-on experience for the participants.

REFERENCES

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol, internet-draft, ietf manet working group, draft-ietf-manet-olsr-08.txt," 2003.

[2] "Freifunk," <http://start.freifunk.net/>, Mar 2009, last visit: 03.2009. [Online]. Available: <http://start.freifunk.net/>

[3] V. Srivastava, A. Hilal, M. S. Thompson, J. N. Chattha, A. B. MacKenzie, and L. A. DaSilva, "Characterizing Mobile Ad Hoc Networks - The MANIAC Challenge Experiment," in *The Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2008)*, 2008.

[4] L. A. DaSilva, A. B. MacKenzie, M. S. Thompson, and E. Q. Baumann, "The MANIAC Challenge: Educational Experiences in Ad Hoc Networking," *IEEE Pervasive Computing*, vol. 8, pp. 7–11, 2009.

[5] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET Simulation Studies: The Incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[6] D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of MANET simulators," in *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*. New York, NY, USA: ACM Press, 2002, pp. 38–43.

[7] J. Macker and R. Lee, "NRL OLSR Routing Protocol Implementation," 2009. [Online]. Available: <http://cs.itd.nrl.navy.mil/work/olsr/>

[8] H. Kazemi, G. Hadjichristofi, and L. A. DaSilva, "MMAN - a monitor for mobile ad hoc networks: design, implementation, and experimental evaluation," in *WiNTECH '08: Proceedings of the third ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. New York, NY, USA: ACM, 2008, pp. 57–64.

[9] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, 2006, pp. 203–216.

[10] R. Axelrod, *The evolution of cooperation*, 1984, vol. 5, no. 3-4.

[11] M. K. AbdelRahman Mohamed, "MANIAC Challenge: The Mongoose Strategy," March 2009. [Online]. Available: <http://www.maniacchallenge.org/AAST.pdf>

[12] M. Caleffi, "MANIAC Challenge: A Diversity Adaptive Approach for Cooperative Behavior," March 2009. [Online]. Available: <http://www.maniacchallenge.org/caleffi.pdf>

[13] M. Günes, F. Juraschek, B. Blywis, Q. Mushtaq, and J. Schiller, "A testbed for next generation wireless networks research," *Special Issue PIK on Mobile Ad-hoc Networks*, vol. IV, pp. 208–212, Oktober-Dezember 2009. [Online]. Available: <http://www.reference-global.com/doi/abs/10.1515/piko.2009.0040>

[14] B. Blywis, M. Günes, F. Juraschek, and J. Schiller, "Trends, advances, and challenges in testbed-based wireless mesh network research," *Mobile Networks and Applications*, vol. 15, pp. 315–329, 2010, 10.1007/s11036-010-0227-9. [Online]. Available: <http://dx.doi.org/10.1007/s11036-010-0227-9>

[15] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 134–146.