# Performance Analysis of Temporally Ordered Routing Algorithm based on IEEE 802.11a

**Erik Weiss*, Guido R. Hiertz*, Bangnan Xu+**
*: Communication Networks, Aachen University, RWTH Kopernikusstr.16, D-52074 Aachen, Germany
+: SSC ENPS (Technologiezentrum), T-Systems, Am Kavalleriesand 3, D-64295 Darmstadt, Germany
Email: erik.weiss@comnets.rwth-aachen.de Phone: +49 241 80 28575 Fax: +49 241 80 22242

***Abstract*—This paper presents a performance analysis of the Temporally Ordered Routing Algorithm (TORA) protocol on top of IEEE 802.11a. IEEE 802.11a offers up to eight different coding schemes. Depending on the coding schemes the ad hoc routing behaviour differs. This paper describes the influence of the coding schemes, reviews the TORA protocol including the link reversal algorithm and presents the combined results. This research focuses on the application of TORA in a scenario where an ad hoc network is connected to the internet using an ad hoc gateway. Special emphasis was spent to a separated investigation of the protocol performance for uplink and downlink routes. The paper presents three enhancements to TORA avoiding and finding loops. Simulation results are presented showing the performance of TORA on top of IEEE 802.11a and describing the feasibility of TORA to be used as routing protocol in ad hoc integration scenarios.**

**Index Terms--** IPonAir, TORA, Link Reversal, Reference Level

## I. INTRODUCING

Nowadays it is increasingly important to be connected to the internet world. The trend is towards the wireless world, providing public access to the Internet via wireless devices at high data rates. Wireless Local Area Networks (WLAN) like IEEE 802.11a work at the 5 GHz band and support transmission rates up to 54 Mbit/s. The high attenuation at 5 GHz limits coverage. Therefore one objective is to extend the coverage by establishing multihop routes. High throughput and limited transmission range make WLAN systems reasonable for areas with a high population density and users with the need for high data rates. Such places are called hotspots like airports or exhibition halls.

### *A. State of the Art*

Multihop connections can expand the fixed infrastructure. To handle mobility and fast topology changes on the network, Ad hoc Routing protocols have been developed. Routing protocols are divided in two groups, the proactive and reactive protocols. Reactive protocols request a route when needed, whereas proactive protocols permanently maintain routes to all network members. Thus, proactive approaches can use the route when requested, therefore minimizing the packet delay. Reactive protocols avoid maintenance of unneeded routes, but to the cost of a higher route discovery and packet delay. Furthermore, hybrid approaches have been developed.

This paper presents the results of our performance evaluation of TORA based on IEEE 802.11a. Most of the previous research focused either on the legacy 802.11 or 802.11b. But small amount of work investigated ad hoc routing over 802.11a. We structured the paper as followed: first we give a brief overview about the IEEE 802.11. Followed by a differentiation of the legacy 802.11 and the 802.11a version, the 5 GHz channel and the different coding schemes are described. The Temporally Ordered Routing Algorithm is described in section three. Section four presents the simulation results. Finally the simulation results are discussed and our considerations are concluded in section five.

## II. IEEE 802.11A MEDIUM ACCESS LAYER INFORMATION

The IEEE 802.11a Medium Access Control (MAC) layer is mainly the same as the MAC layer of 802.11b and the legacy 802.11. The main differences to 802.11a are the transmission modes (Table 1). 802.11a can chose between eight coding schemes, so called *"PhyModes"* (cf. Table 1). IEEE 802.11 uses a distributed MAC protocol, the Distributed Coordination Function (DCF). The 802.11 standard also defines a Point Coordination Function (PCF) but no vendor has implemented it. The DCF is based on carrier sense multiple access with collision avoidance (CSMA/CA). For wireless mobile nodes (MN) it is not possible to monitor the air interface while sending. Hence, the DCF uses backoff and request-/ clear to send (RTS/CTS) mechanisms to avoid collisions. Details of the IEEE 802.11a MAC protocol are shown in [1].

### *A. IEEE 802.11a Transmission modes*

IEEE 802.11a offers eight different transmission modes. The standard itself does not specify any rules for selecting the PHY mode [1].

| Data rate (Mbit/s) | Modulation | Coding rate (R) | Data Bits per Symbol |
|---|---|---|---|
| 6 | BPSK | 1/2 | 24 |
| 9 | BPSK | 3/4 | 36 |
| 12 | QPSK | 1/2 | 48 |
| 18 | QPSK | 3/4 | 72 |
| 24 | 16-QAM | 1/2 | 96 |
| 36 | 16-QAM | 3/4 | 144 |
| 48 | 64-QAM | 2/3 | 192 |
| 54 | 64-QAM | 3/4 | 216 |

**Table 1: Mode Dependent Parameters**

Fig. 1 presents the Packet Error Rate (PER) versus Carrier to Interference (C/I). Higher transmission modes are capable to deliver higher data rates, but nevertheless, they also need a remarkable higher C/I. In Table 1 the available modes are listed together with the maximum data rate and the bits per OFDM symbol.

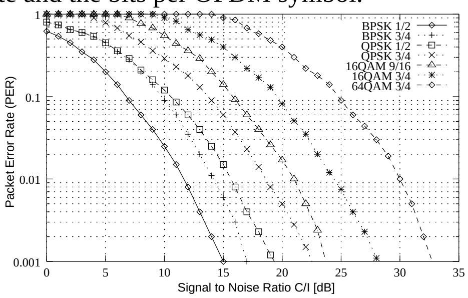


**Fig. 1: Packet Error Rate versus C/I**

Due to the dependence between C/I and useable transmission mode, IEEE 802.11a allows to change the transmission mode when the channel quality is decreasing. Decreasing the channel quality implies several reasons. The IEEE 802.11a system offers the opportunity to choose an appropriate PhyMode for every connection and every data packet.

## III. TORA: TEMPORALLY ORDERED ROUTING ALGORITHM

The Temporally Ordered Routing Algorithm is an adaptive distributed routing algorithm for multihop ad hoc networks. It was intentionally build for fast changing network topologies. The protocol is based on the link reversal concept described later in section C. TORA uses destination oriented routing information that is already available at each node. Nodes only need to know their one-hop neighbourhood. Based on the neighbour information TORA creates independently local routing information for each destination node. The destination oriented routing principle allows reactive, proactive and combined concepts. Furthermore TORA was drafted to be able using multiple routes and absence of loops. TORA uses four different control message types, *Query* (QRY) for route discovery, *Update* (UPD) to update the routing structure and height tables, *Clear* (CLR) to delete invalid routes and *Optimization* (OPT) for route optimization.

### A. TORA Principle

TORA associates for each destination a metric to each node; this metric can be interpreted as height *H(i)* of the node *i*. The height is composed of five different parameters. Equation 1 shows the respective quintuple.

$$H(i) = (\underbrace{tau(i), oid(i), r(i)}_{\text{Reference Level}}, \underbrace{delta(i), i}_{\text{Offset}}) \qquad (1)$$

The first three parameters define the reference level and the other two the offset. The parameters have the following meaning.

- **tau:** Time of the last reference level update
- **oid:** Identification (ID) of the node who defined the last reference level
- **r:** Flag if the reference level was reflected
- **delta:** To separate nodes with equal reference levels
- **i:** Unique node ID

It is required that all nodes are synchronised to the same clock, since the time is part of the metric. This might be achieved using the Global Positioning System (GPS) for instances. The parameters in Eq. (1) are rated in decreased order form the left hand to the right hand side.

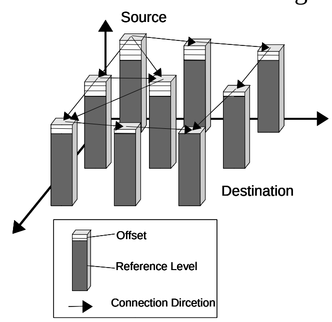


**Fig. 2: TORA Principle**

Fig. 2 visualizes the principle; any data packet transmission is always routed from a higher to a lower height-node. Since the destination is the node with the lowest height the packets are flowing down to the destination.

### B. Internet MANET Encapsulation Protocole (IMEP)

TORA is a pure routing protocol and covers only the routing functionality. It base upon mechanisms and services contributed from lower layer protocols. The required functions are as follows:

- Neighbourhood monitoring and discovery
- Reliable and successive transport of control messages
- Address mapping of IP to MAC addresses
- Authentication

IMEP has been developed to support TORA with the required functions [6].

### C. Link Reversal Algorithm

The link reversal algorithm for routing in wireless networks was first presented by Gafni and Bertsekas [6] and bases on modelling the wireless network as a directed acyclic graph (DAG). The height of a node has to be chosen under the condition that a totally lexicographical adjustment of the nodes is possible. Thus, the routing directions for each adjacent node pair are defined from 'higher' to the 'lower' node. The graph is called, destination oriented in the case that each node can reach the respective destination node.

Transforming a graph to a destination oriented graph is the main task of the link reversal algorithm. Two link reversal algorithms have been developed, a partial and a full reversal approach. Fig. 3 describes both approaches.

*Full Reversal Method*

With each iteration all uplink (directed to a node itself) edges are turned for nodes that have no downlink edges (directed to neighbour nodes), independently if the edge has been turned already or not.

*Partial Reversal Method*

Different from to the full reversal method a node under partial reversal method turns only those edges where the respective neighbour nodes have not yet done the partial reversal. In case all neighbours to a node have already done a partial reversal, all edges must be reversed. Both algorithms are loop free and in case the network is not partitioned, the number of iterations required is limited. If the network consists of several parts, the basic reversal algorithm is not stable and does not terminate. Fig. 3 depicts the principle. On the left the Full Reversal and on the right the Partial Reversal Method is shown. Both methods start after the link between node (A, B) has failed. The first iteration reverses the edge from node A to node E. Full reversal continuous reversing the edge (E, D) and (A, E) again. The partial reversal method only turns the edges (E, A), (D, E) and terminates after the second iteration.

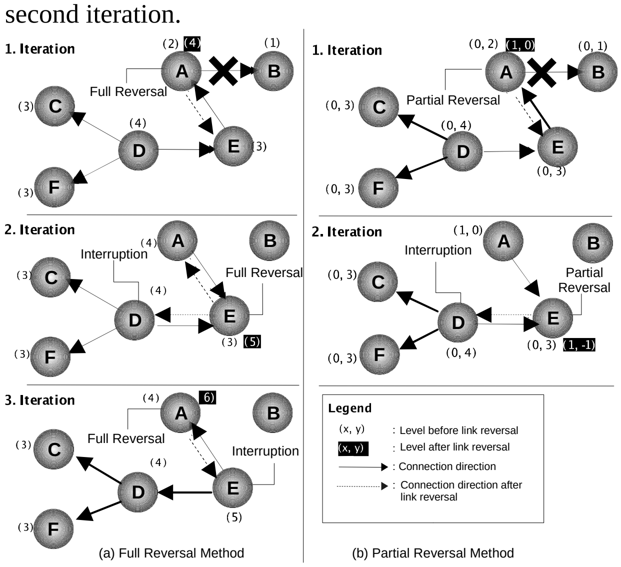


**Fig. 3: Full vs. Partial Reversal Method**

The Full Reversal method continues in the third iteration, since node A has turned its former downlink and consequently node A must return this edge again. Fig. 3 clearly visualizes that partial reversal performs more efficient than full reversal.

TORA uses for maintaining the routes an approach similar to the partial reversals method. TORA performs the described edge reversal by choosing and updating the node height. Assuming in a destination oriented DAG one node has lost its last downlink (outgoing) link. The basic idea is that this node $K_i$ increases its reference level $R_{new}$ to be higher than all its neighbour nodes. This node represents a maximum height and all edge directions based on the reference level, with increasing the level all incoming edges are turned. Subsequently adjacent nodes might lose their former downlink connection. Each neighbour node that has lost its former outgoing edge does a partial reversal on edges to its adjacent nodes. To turn the edges a node chooses a new reference level based on the reference level $R_{new}$ of its neighbour node that has already adapted its edge to it. Hence, the new defined reference level $R_{new}$ is propagated through the network. The reversal stops at nodes that still have at least one outgoing link. Choosing a new reference level is equal to a local ‘route request’ for a particular destination. When a node sets a new reference level, it initiates a new search for routes to respective destination. When the search process reaches a node $K_j$ at the network boarder and all nodes adjacent to $K_j$ have already turned their edges and have adapted their reference level, node $K_j$ sets the r flag and thus defines an intermediate level according to $R_{new}$. This intermediate level sows, by means of the flag indication that the network boarder has reflected the search process. If all boarder nodes are reflecting the reference level $R_{new}$, this denotes a failure of the search process. The initiator node detects the route request failure after all neighbour nodes having reported a reflected reference level to it. The destination then does not exist or the network is partitioned. If this situation does appear the routing entries created form the previously route search have to be deleted, using the DEL message. Fig. 4 presents an example of the TORA protocol. As already mentioned a node connectivity graph is build and maintained by updating and distributing the node height values. Whenever a node changes its height, adjacent nodes must be informed using an UPD message. This may cause a node loosing the former downlink edge.

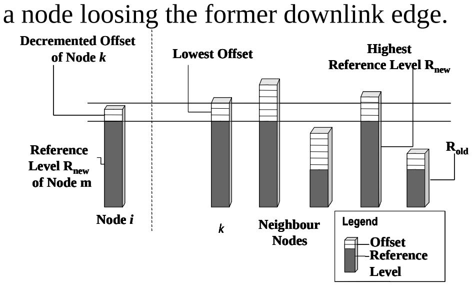


**Fig. 4: Choosing the height based on the reference level $R_{new}$ from the adjacent node and on a decremented offset from node *k***

Hence, the informed nodes themselves have to update their heights and to propagate this information further. Node i chooses its new level according to its highest neighbour reference level and the offset in respect to the smallest neighbour offset (from those with new reference level) decremented by one (cf. Fig. 4). Thus, the node partly turns the edge directions. This behaviour ensures that only edge directions are turned at nodes whose adjacent nodes have not done a partial reversal so far. Edges directed to nodes that have already done the partial reversal are kept unchanged. The new resulting height of a node has to be transmitted to all neighbour nodes.

## D. TORA: Extensions

The TORA protocol has been implemented based on [5]. Further extensions have been made to enhance the protocol performance.

*Intermediate queuing of data packets*

Since all nodes are moving there might be a short period where a route is not available, to overcome these gaps, and to increase the protocol performance, packet queues at each node are implemented to store the data packets for a short time.

*Loop Avoidance*

Although, the protocol was developed to be free of loops, a simulative analysis has shown that in a wireless environment, were reliable packet delivery is hardly to realize, many loops have been created. TORA works perfect as long as all control messages are immediately delivered. Unfortunately that is not always possible in a wireless network. When transmission error occurs for instances an UPD is lost, the route is not adapted properly. Fig. 5 depicts how control message loss may create loops (b). The UPD from F to node B is lost.

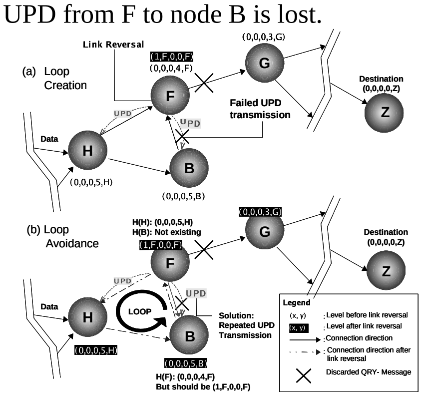


**Fig. 5: Loop Formation and Avoidance**

The IMEP protocol entity of node F reports TORA that the neighbour F is not longer reachable, thus TORA discards all routing information in the neighbourhood table for node B and sets the next hop to F. Fig. 5 (b) presents the way packets would be transmitted when arriving at this stage, a loop is formed.

To avoid such a situation TORA has been extended. If a node receives a data packet it checks first whether the node is known as neighbour or not. Is the sending node unknown IMEP informs TORA about a new detected neighbour and about the way the node has been detected. Has the detection been based on a received data packet, TORA knows that the routing information contain inconsistencies. To fix the problem the node sends again an UPD message containing its height, thereupon the receiver changes its height and the loop is eliminated.

*Loop Recognition*

Loop avoidance solves only the UPD loses in the directed neighbourhood, however within a real wireless environment many reasons for lost or delayed control messages exits. For instances under high traffic conditions, it is common that control messages are send with a latency of several hundred ms. In the meantime the routing information are not consistent at each node. This is the main reason of loop forming. To fix this problem a windowing mechanism has been implemented. Each node monitors passing data packets; and stores a session and sequence number of the last $t$ packets. When a node receives a packet it checks if the packet passed the node already. If it arrives a second time the packet is discarded. The efficiency of this approach bases on the number of remembered packets $t$. In a real TORA implementation a trade-off between window size and used memory has to be found.

## IV. Performance Evaluation

To measure the performance of TORA on top of IEEE 802.11a, the protocol has been implemented and investigated using an event driven simulator. We investigated the following scenarios.

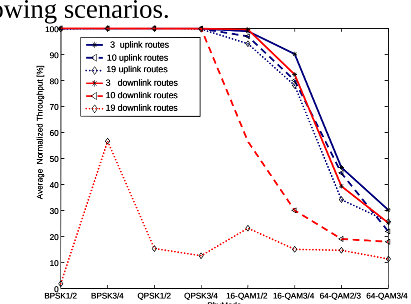


**Fig. 6: Uplink vs. downlink normalized Throughput**

The goal was to analyse the performance separated for an increased number of uplink or downlink routes. The scenario is a square of 80x80 m containing 19 mobile stations plus 1 station operating as a gateway to the internet. All routes either start at the gateway (downlink) or end at the gateway (uplink). The gateway node has been placed at the positions in the upper third. All mobile nodes are moving following the random-way-point mobility model [3] with a speed chosen to 1 m/s (pedestrian). Different numbers of routes have been investigated starting from 3 routes, up to 19 routes. Each route has been burdened with 128 kbit/s, constant bit rate (CBR) and a packet size of 512byte. This paper presents the achievable throughput, the measured overhead and the number of route breakages.

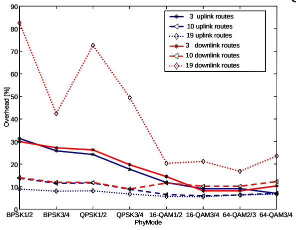


**Fig. 7: Uplink vs. downlink overhead percentage**

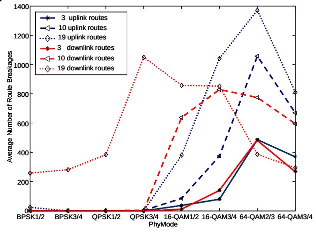


**Fig. 8: Uplink vs. downlink average number of route breaks**

Fig. 6 presents the achievable throughput versus different PhyModes offered by IEEE 802.11a. Downlink connections perform much better, than the uplink connection. This behaviour can be observed by most routing protocols, since all routes are looking for the same destination and therefore the routing information are highly distributed. Downlink connection on the opposite, each route has different destinations hence the routing information has to be gathered separately for each destination. With increasing number of routes and also increasing offered traffic, it can be observed, that high rated PhyModes are not able to carry the traffic. With 19 routes to different destinations, no PhyMode is able to establish a reliable multihop connection. A particular situation can be observed with the lowest PhyMode BPSK ½. This PhyMode supports long transmission distances, thus most destinations can be reached by one hop. On the other side this also means that all node are interfered form all other nodes, hence the wireless medium must be shared between 20 stations and no spatial reuse can be applied. Furthermore using BPSK ½ means transmitting very long packets since the coding efficiency is low. With BPSK ¾ one can observe that for 19 routes the throughput increases, but further PhyMode increasing can not improve the throughput. Comparing the maximum end-to-end throughput (Fig. 6) with the measured overhead in Fig. 7 showed as expected the overhead for downlink routes is always higher than for uplink routes. For the simulation campaign with 19 routes in downlink direction similar effects than in Fig. 6 can be seen. With BPSK ½ the medium is congested, TORA is not able to establish the routes. 80% of the network traffic is overhead and only 20 % are data.

Fig. 8 shows the average number of monitored route breaks. With the low rated PhyModes the routes are more Stable. A very small amount of breakages occur, except the 19 routes campaign. As soon as the PhyMode is higher rated the number of breakages increase tremendous.

## V. Conclusion

This paper reviews the TORA protocol in combination with IMEP and the impact of the coding potentials offered by IEEE 802.11a. TORA has been enhanced with loop avoidance and loop recognition however a loop creation cannot be denied in general. This is due to typical behaviour in wireless environments. The paper present simulation results focussing on different coding schemes and distinguishing the behaviour per PhyMode. Also the differences between uplink and downlink connections have been addressed. TORA shows a sufficient performance in scenarios where most routes are going towards a common node e.g. access point or access router, the behaviour is promising and might enable TORA to run in large ad hoc networks for the uplink direction. Investigations of larger ad hoc networks are our next steps. Unfortunately the downlink direction performance is weak and for downlink connections TORA is not able to compete with other routing protocols. The developed enhancements increased the performance but the overall results for downlink connections shows that TORA is too weak. TORA might be applicable in a scenario where many people are transmitting towards one common destination node. A gateway or access point might be such a node; TORA could be applied only of the uplink and some other routing protocol for the downlink. This separation depends on the feasibility of two different protocols to share their information to increase the combined uplink and downlink performance.

## VI. Acknowledgement

This work was supported by T-Systems, Deutsche Telekom and the German research project IPonAir, funded by the German Ministry for Education and research (BMBF). The authors would like to thank the members of the project for the valuable discussion.

## References

[1] IEEE LAN/WAN Standards Committee, "*Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in 5 Ghz Band*," IEEE Std 802.11a-1999, IEEE, New York, Nov. 1999.

[2] C. E. Perkins. *Ad Hoc Networks*, USA, Addison-Wesley, December 2000.

[3] Bettstetter C., Hartenstein H., Pérez-Costa X. *Stochastic Properties of the Random Waypoint Mobility Model.* ACM/Kluwer Wireless Network, Special Issue on Modelling & Analysis of Mobile Networks, March 2003.

[4] E. Gafni, D. Bertsekas. *Distributed algorithms for generating loop-free routes in networks with frequently changing topology.* IEEE Transactions on communications, Vol. Com-29, No1. 1981.

[5] V. Park, S. Corson. *Temporally-Ordered Routing Algorithm (TORA) Version 1*. IETS Internet darft (draft-ietf-manet-tora-spec-04.txt), July 2001.

[6] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, A. Qayyum. *An Internet MANET Encapsulation Protocole (IMEP) Spécification*. IETF Internet draft (draft-iets-manet-imep-spec-01.txt), August 1999.