

# Ingress Throttling and its Applications in IEEE 802.11 Based MANETs

Evgeny Osipov and Christian Tschudin

Original scientific paper

**Abstract**—One of the stumbling blocks which prevents multihop ad hoc networks from wide deployment is a known problem of unfair bandwidth distribution between competing data sessions. Practically, a combination of the current IEEE 802.11 technology and the standard TCP protocol allows only a couple of up to three hops connections co-exist simultaneously while providing fairly stable and acceptable service to end users. In this article we describe ingress throttling, a resource protection layer which restores fair bandwidth sharing between plain TCP as well as arbitrary UDP sources. We summarize analytical as well as simulation studies of our solution and demonstrate its applications for determining scaling limits of ad hoc specific simulations and evaluating the effect of ad hoc routing on performance of data communications.

## I. INTRODUCTION

The standard TCP protocol is known to have poor and unstable performance over wireless links, especially in the wireless multihop case. One of the key problems is an extremely unfair distribution of network capacity between competing sessions due to *TCP capture* [1]. As it is experimentally shown in [2] the IEEE 802.11 MAC protocol being unable to handle collisions more than one hop away results in a situation where few lucky TCP sessions occupy the available bandwidth pushing the competing connections in a continuous slow start phase. The problem is linked to the inability of TCP's congestion control to differentiate the packets losses due to radio interferences from those induced by network congestion [3]. Fig. 1, generated from simulations, illustrates the degree of unfairness between multiple TCP sessions in wireless networks with static routing and no mobility for different numbers of flows and hops. We argue that this issue needs to be solved by an architectural approach.

### A. Contribution of the article

In this article we provide an analysis of the unfairness problem and present a *resource protective* network architecture for wireless ad hoc networks that solves the unfairness problem. The formal part of the architecture relies on the *max-min* fairness model of the wireline Internet which we adapt

Manuscript received October 20, 2006 and revised November 22, 2006. This paper was presented in part at the 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN) 2006.

E. Osipov is with the Department of Wireless Networks, RWTH Aachen University, Germany, (e-mail: Evgeny.Osipov@mobnets.rwth-aachen.de).

Ch. Tschudin is with the Computer Science Department, University of Basel, Switzerland.

to the specifics of multihop wireless communications. On the implementation side we present a corresponding *ingress rate throttling* scheme that enforces the model and solves the unfairness problem. The major improvement we achieve by throttling the output rate at ingress nodes is an increase in total network throughput and almost perfect fairness.

We also present two useful applications of our ingress throttling approach for dimensioning of experimental scenarios and evaluating the effect of ad hoc routing on performance of data communications. In the first case we derive scaling limits of ad hoc specific simulations in terms of numbers of simultaneously active data sessions in a given geographical area. Beyond these limits the per-flow best case performance does not satisfy even minimal user expectations. In the second case we consider reactive ad hoc routing schemes operating in the path maintenance phase. We study the effect of *four* distinct routing traffic patterns on the quality of ongoing TCP sessions. For each considered routing scheme we estimate the “equivalent routing load”.

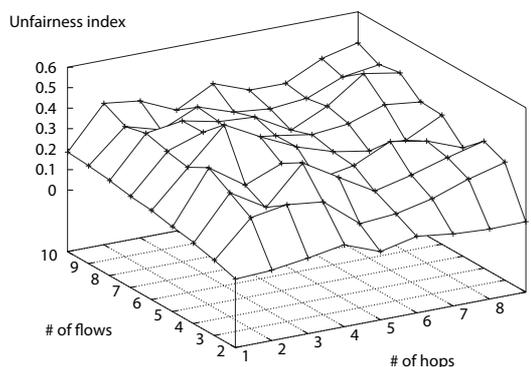


Fig. 1. TCP unfairness index (simulations). See Fig. 3 for the topologies.

### B. Outline

We develop the topic as follows. We first summarize the design of the ingress throttling architecture in Section II. Selected performance results are reported in Section III where we also discuss the usage of our solution for dimensioning of experimental scenarios. Our approach for evaluating ad hoc routing protocols follows in Section IV. We summarize the material and conclude the article in Section V.

## II. THE INGRESS THROTTLING ARCHITECTURE

The design flow followed to arrive at our architecture is depicted in Fig. 2<sup>1</sup>. In a nutshell our solution adapts the max-min fairness model from the wireline Internet. Recall that this model governs the way how limited network resources (capacity of communication links) are distributed between multiple competing data sessions. In essence the model says that “nobody can benefit at the expense of another”.

Attempting to reflect the wireline fairness model to the case of multihop wireless networks we understood that the major concepts such as the *source*, the *link*, the *rate* of sources and the *capacity* of the links are not suitable for wireless networks. We propose entities which serve as substitutes for the corresponding terms in multihop wireline networks and shift the focus from the wireline-specific *link-capacity* domain to the wireless-specific *space-load* domain (see label “a” in Fig. 2): This leads to the new definitions of *bottleneck region* and the *boundary load* which correspond to the wireline terms of “bottleneck link” and “capacity”.

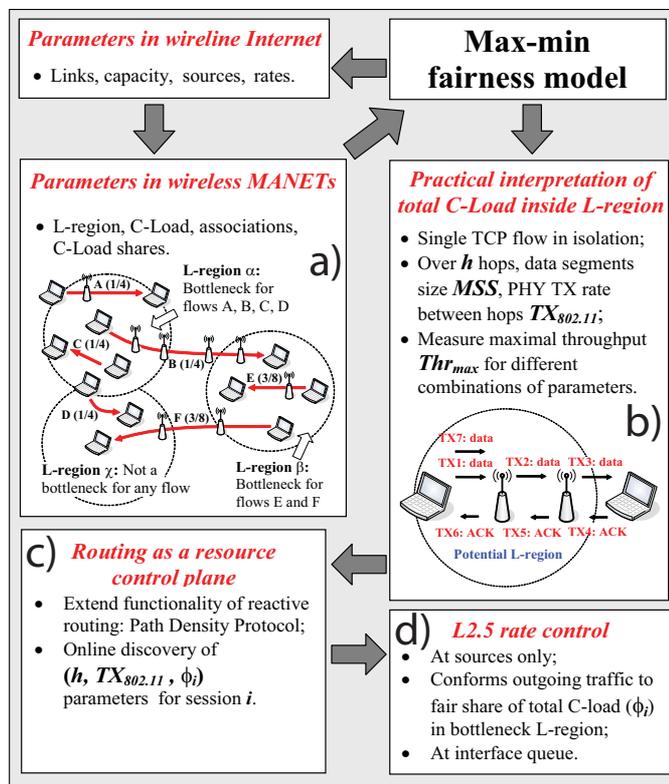


Fig. 2. Design flow for the ingress throttling architecture.

In a second step (label “b” in Fig. 2), we derive an ingress rate limit which ensures that the sum of the loads produced by all data flows inside the bottleneck region does not exceed the boundary load. For this we propose an original way of characterizing capacity of a geographical region based on

<sup>1</sup>For the sake of clarity in this and the subsequent figures nodes that participate only in relaying traffic for other users are indicated by wireless relay symbols, while the source and the destination nodes are shown with laptop symbols.

maximally achievable throughput of a single multihop TCP session.

As a third step, we use routing as a resource control plane (label “c” in Fig. 2): We extend the functionality of routing protocols to deliver the fair shares of region’s load (which depends on the current state of the network) to the sources of competing data sessions. This information is used by these ingress nodes to compute the local rate limit. We apply the derived rate limit to configure a scheduler at the interface queue and shape the outgoing traffic accordingly before it enters the transmission queue at the MAC layer (label “d” in Fig. 2). With this scheme none of the TCP sessions is able to benefit from temporal weaknesses of the competitors by capturing the transmission capacity.

### A. Max-min fairness model for wireless networks

In this section we present the most salient features of the adapted max-min fairness framework and overview the key principles of the ingress throttling architecture. For more details we refer the reader to [4]. We begin with the definition of the basic terminology. We argue that these definitions should make the difference between wireline and wireless network fairness optimization more clear.

1) *From wireline “link” to wireless “L-region”*: In wireless networks (including IEEE 802.11), the term “link” as it is used in classical routing and graph theory based work is misleading since the radio signal for a given packet transmission propagates in a geographical region of a certain size. We define the *L-region* as the area around a wireless node transmitting or receiving data of at least one end-to-end data flow that corresponds to the size of the 1 Mb/s transmission range of an IEEE 802.11 radio transmitter.

We say that a node carrying traffic of a specific end-to-end data flow belongs to the particular L-region if it is able to communicate with the central node of that L-region with the base IEEE 802.11 transmission rate of 1 Mb/s.

The rationale for defining L-region as 1 Mb/s transmission range is twofold. Firstly, in reality the shape of L-region is complex and is not an ideal circle as Fig. 2 shows. However, defining the L-region as IEEE 802.11 basic rate transmission range we do not assume any specific radio propagation model and allow for an arbitrary shape of the L-region. Secondly, we need means of communication between nodes carrying traffic of competing connections. With such definition two data flows located outside the range of assured data reception have a possibility to communicate with each other through the central node of an L-region in between. This is illustrated in Fig. 2 (label “a”) where flows A and D can discover the presence of each other through nodes carrying the traffic of flow B in L-region  $\alpha$ .

2) *From wireline “sources” to wireless “associations”*: We define a *source-destination association* as the set of nodes that forward packets of a specific TCP flow, as well as its source and destination node. We say that a node of a specific association belongs to the particular L-region if it is able to communicate with the central node of that L-region with the base IEEE 802.11 transmission rate of 1 Mb/s.

3) *From wireline “rate” and “capacity” to wireless “load share” and “boundary load”*: Our definition of the L-region makes the terms “rate” of sources and “capacity” of the link in their conventional sense meaningless in multihop wireless networks. The major reason for this is the multiple transmission rates of the IEEE 802.11 based devices available at the physical layer (e.g. 1 Mb/s, 2 Mb/s, 5.5 Mb/s, 11 Mb/s for the IEEE 802.11b). Since nodes inside the L-region may use any of the available physical layer transmission rates, it would be a complex task to define a formal mapping of each rate to a single entity referring to the capacity of a geographical region.

As a resource to share within the L-region we therefore define the *load* which competing associations *generate* or *consume* (require) inside the L-region. We call this term the *conserved load* (C-load).

We define a *C-load share* ( $\phi$ ) to be the analogue of the wireline “rate”: It is the fraction of the total C-load that a particular flow generates or consumes inside the L-region.

4) *Definition of max-min fairness in the space-load domain*: With the above defined substitutes for *source*, *link*, *rate* of sources and *capacity* of the links, we formulate the *space-load* max-min fairness as follows. A feasible allocation of *C-load shares* for the competing *associations* is “max-min fair” if and only if an increase of any *C-load share* within the domain of feasible allocations must be at the cost of a decrease of some already smaller *C-load share*. This is achieved when every association shares a bottleneck *L-region*.

An L-region is said to be a *bottleneck* for a particular association if the L-region is saturated (e.g. its C-load is fully distributed between the competing associations) and the C-load share of the considered association is larger than or equal to the C-load shares of other associations which share this L-region. This concept is illustrated in Fig. 2 (label “a”) where flows A, B, C and D receive C-load shares 1/4 in bottleneck L-region  $\alpha$  and flows E and F get C-load shares 3/8 in bottleneck L-region  $\beta$ .

5) *TCP throughput as a reference to the C-load*: In order to manage the bandwidth resources, we need to find the condition under which every node of an association tends to generate maximal load inside a geographical region. In the wireline Internet this condition relates to the *bandwidth-delay product* – the amount of traffic that the entire path can accommodate. In the case of a wireline network, TCP is able to estimate the bandwidth-delay product: A single TCP flow in steady state is a perfect estimator of the available bandwidth in the network. In the wireless case, however, TCP fails to achieve steady state because of self-induced interference: Only below a certain threshold  $Thr_{max}$  will TCP show stable performances.

## B. Determining the throttling rate

Let  $Thr_{max}(h, MSS, TX_{802.11})$  denote the maximal throughput achieved by a single TCP connection in a network that is free from competing data sessions.  $h$  is the number of hops traversed by the flow,  $TX_{802.11}$  is the transmission rate at the physical layer and  $MSS$  is the size of data segments generated by the source.

The value  $Thr_{max}$  is source-specific and refers to the C-load in the flow’s bottleneck region. It can either be analytically determined, which would require a formal treatment of the TCP and IEEE 802.11 MAC protocols. Or, this value can be experimentally measured through simulations for all combinations of input parameters, which is what we have done.

1) *The ingress throttling formula*: Assume that we now know the boundary C-load of a single flow: If several flows share the same L-region, they must reduce their data rate according to the share factor  $\phi_{bottleneck}$ :

$$r_i^{ingress} \leq Thr_{max}(h, MSS, TX_{802.11}) \cdot \phi_i^{bottleneck}. \quad (1)$$

This formula asserts that bandwidth resources are allocated according to max-min fairness such that TCP sessions in a wireless network achieve stable operation.

2) *Enforcing rate throttling at the ingress nodes*: In our *space-load* fairness framework we want that all ingress nodes conform to the formula above. To this end, each ingress node needs 1) to choose the correct value of the maximal throughput  $Thr_{max}$ , and 2) to obtain the fair C-load share for this connection in the network. By modifying ordinary routing protocols, all required values ( $h, \phi_i^{bottleneck}, TX_{802.11}$ ) can be obtained at run time; In [5] we presented such an extension and do not discuss routing issues further in this article.

Having obtained the value for the fair bottleneck capacity and corresponding rate  $r_i^{ingress}$ , we can set the delay parameter of a packet scheduler as  $\Delta = \frac{MSS_i}{r_i^{ingress}(h, MSS, TX_{802.11})}$ . Data traffic that is shaped according to this scheduler can be of any kind, not just TCP traffic: TCP was only used to determine the maximal load, although TCP at the same time is the first protocol to benefit from respecting this rate limit because of the gained stability.

## C. Existing approaches towards enhancing TCP performance

The discovered incompatibility of the standard TCP protocol with wireless media triggered numerous studies targeting an improvement of the situation. Overall they can be classified into two major categories: Improving the performance of IEEE 802.11 MAC and link layer, and adaptation of TCP to wireless environments. A description of all or even the major representatives in each class would be a too ambitious task for this article. We refer to extensive surveys of the existing approaches in [6], [7] and references therein.

The studies in the first class apply the existing quality of service enhancing schemes from the wireline Internet to multihop wireless networks. The results of these approaches are distributed algorithms for exchanging the status of local transmission queues at the link layer and scheduling information in the transmission range of every wireless node. A typical representative of these approaches is the distributed version of RED algorithm [8].

The approaches dealing with adapting the TCP’s congestion control mechanism to the specifics of the wireless transmission medium can be divided into several subcategories. The first

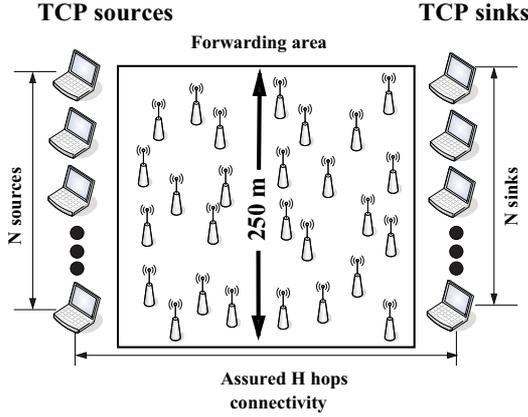


Fig. 3. Considered network topology.

subcategory aims at creating means for TCP senders to distinguish between the packet losses caused by congestion, radio interference induced bit errors and unavailability of routes to the destination. A typical representative of these approaches is ATCP [9]. The second type of proposals is receiver-oriented. These approaches try to steer the behavior of senders by means of smart acknowledgment generation techniques. TCP-Eifel [10] is a typical representative of this type of schemes.

Our solution is different in that it does not attempt to improve TCP's congestion control nor the functionality of the MAC layer, neither do we propose a scheme to discriminate the packet losses. Rather, we create a smart session-oriented engine very close to the interface queue i.e., at layer 2.5: This thin *resource protection* layer is aware of the consequences of the uncontrolled TCP transmissions over multihop wireless networks and adjusts the behavior of the outgoing traffic to maximize the benefits both for the locally originated session as well as all competing (TCP) flows in the network.

#### D. Summary of the ingress throttling

In this section we overviewed the key aspects of the max-min fairness model adapted to the specifics of wireless communications. In particular we presented new entities called the L-region, the C-load share and the boundary C-load, which serve as substitutes for the corresponding terms the link, the rate and the capacity in multihop wireline networks.

We summarized practical issues of the space-load fairness model enforcement in MANETs. Taking the ideal throughput of a single TCP session as a reference to the boundary load of the L-regions we computed the limit on the ingress transmission rates, which ensures that the total load from multiple TCP connections inside the bottleneck L-region does not overflow the boundary load.

### III. EXAMPLES OF PERFORMANCE GAINS DUE TO INGRESS THROTTLING AND DISCUSSION

For our simulation experiments we use the topology in Fig. 3. There we have three geographical areas with sources of TCP sessions, forwarder nodes and sink nodes. The forwarding

area begins and ends one wireless hop away from the area with the sources and the sinks, respectively. The size of the forwarding area ensures  $H$  hops communications between each source and sink. The number of nodes in the forwarding area assures potential connectivity for each source-destination pair. In this topology we are able to vary three parameters: The number of competing TCP sessions, the hop length of flows and the number of nodes in the network.

We used the ns-2 network simulator<sup>2</sup> (version 2.27) and TCP New Reno as the most popular variant of the protocol. In all simulations we used continuous FTP traffic from all sources. We set the value of the TCP maximum segment size (MSS) to 600 B; RTS/CTS handshake is disabled; Other ns-2.27 parameters have default values. In the simulations presented in this section the routes for all flows are statically assigned prior to the data transmissions. As all TCP flows started we allow a warm-up period of 12 seconds to exclude initial traffic fluctuations from the measurements. The duration of all simulations is 120 seconds.

In all experiments we assume that the information about the bottleneck C-load shares, the physical layer transmission rates on the path and the estimates of the ideal throughput for the flows with corresponding parameters is available at sources of TCP flows. Since the main purpose of the experiments is a functional assessment of our solution and not an auto-configuration of the used parameters for our ingress throttling scheme, we configure the delay parameter of the scheduler to the MAC layer transmission queue prior to the start of the experiments.

1) *Used performance metrics:* We assess TCP performance using the following set of metrics:

- Combined (total) TCP throughput of all existing in the network TCP flows. Denote this metric  $Thr_{tot}$ ;
- Unfairness index  $u$ : It is the normalized distance (2) of the actual throughput of each flow from the corresponding optimal value:

$$u = \frac{\sqrt{\sum_{i=1..N} (Thr_{opt_i} - Thr_{act_i})^2}}{\sqrt{\sum Thr_{opt_i}^2}}. \quad (2)$$

In this formula  $Thr_{opt_i}$  is the ideal throughput of flow  $i$  obtained under fair share of the network capacity. In order to compute this value we apply the fair share of the C-load for a particular flow in its bottleneck ( $\phi_{bottleneck}^i$ ) computed for a particular scenario to the flow's throughput obtained when running alone in the network.  $Thr_{act_i}$  is the actual throughput of the same flow achieved while competing with other flows. This index reflects the degree of efficiency of actual capacity allocation with respect to optimal fair values. The closer the value of the index to 0 the more fair and efficient the system performs.

#### A. Flows with equal initial opportunities

This experiment we perform on a set of TCP flows with equal transmission potentials all crossing the common bot-

<sup>2</sup>[Online]. Available: <http://www.isi.edu/nsnam/ns/>

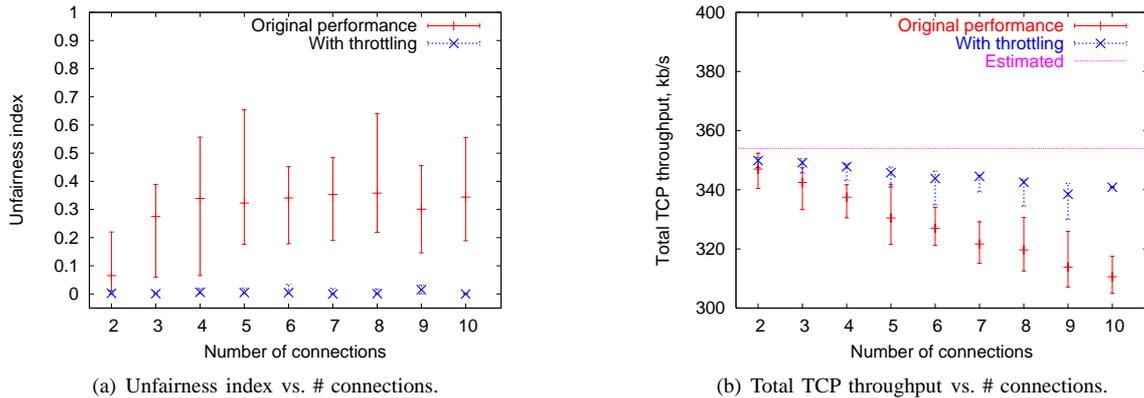


Fig. 4. Improvements of TCP performance in multihop ad hoc networks achieved with the ingress throttling.

tleneck L-region. In the network depicted in Fig. 3 we set the number of hops for all flows  $H = 3$ . Assuming the transmission range of a node equals 250 m (the default value in ns-2), it is possible to show that with these settings the majority of network nodes are located inside the single bottleneck L-region which area coincides with the forwarding area in the figure. This network allows us verify the validity of our hypothesis to use the maximal throughput of a single TCP connection as a reference to the boundary C-load inside an L-region and to perform a scalability test. In the first case indeed, if it is a wrong hypothesis then the total TCP throughput of all TCP connections running without shaping will be higher than this value. As for the scalability test, varying the number of connections inside the L-region we will monitor the dynamics of the used performance metrics depending on the number of competing connections. In order to make the flows alike we set the data transmission rate of all devices to 2 Mb/s and all flows use equal TCP maximum segment size. We increase the number of simultaneously active flows in this experiment from 2 to 9 performing 30 simulations for each value.

Fig. 4 shows the dynamics of the unfairness index (2) and the combined TCP throughput in the network  $Thr_{tot}$  for different numbers of the competing connections inside the common L-region. All graphs show the mean values and the range between the minimal and maximal values of the corresponding metrics over 30 simulation runs. In each run we seeded the random number generator of ns-2 randomly.

In the case where no shaping is done at sources TCP flows can transmit as much traffic as they can. As we observe from Fig. 4(b), the total TCP throughput in this case is always lower than the throughput of a single three hop session (the straight line in the figure marked as “Estimated”). By this we confirm our hypothesis to consider TCP throughput of a single TCP flow in isolation as a reference to the boundary C-load of the L-region.

From Fig. 4 we see a significant improvement of the quality of TCP communications in the case of enabled ingress throttling. Under our scheme we achieve an increase in the total TCP throughput and almost perfect fairness which also indicates an increase of individual throughputs.

It is important to point out the stability of the performance

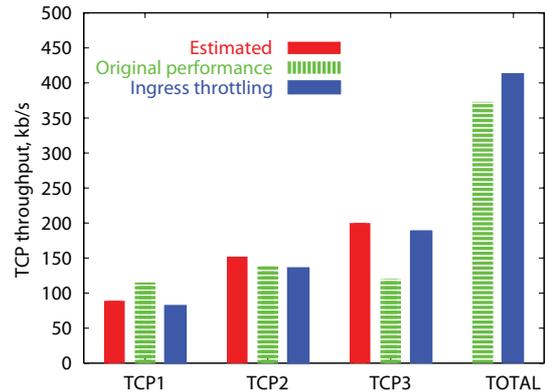


Fig. 5. Per-session and combined TCP throughputs in the experiment with variable physical layer transmission rates.

metrics. When the ingress throttling is enabled the network consistently shows much better performance in all simulation runs. At the same time the service offered by the plain combination of TCP and IEEE 802.11 MAC is highly variable and to a large extent unpredictable.

### B. Flows with variable physical layer transmission rates

In this experiment we assess a very important aspect of the validity of our theoretical findings in a network where the competing flows might use any of the available transmission rates of the IEEE 802.11b physical layer. This time there are three competing four hops TCP connections which use correspondingly 2, 5.5 and 11 Mb/s transmission rate between the hops. All connections use  $MSS = 600$  B. The performance results are shown in Fig. 5.

As for the original network performance depicted in the figure by the middle bars in each group, we see that flow *TCP1*, which uses the slowest transmission rate, achieves higher throughput than the estimated value under fair allocation of C-load shares. This is a natural situation known as performance anomaly of the IEEE 802.11 MAC [11]. In this phenomenon the MAC protocol attempting to provide max-min allocation of the transmission rates at any node favors transmissions with slower rates by this degrading the performance of potentially

faster transmissions.

Enabling our ingress throttling we restrict this flow to use only its share of the C-load in the L-region. By doing so the resulting per-flow throughputs are proportional to the underlying transmission rates. As a result we also observe an increase in total TCP throughput in the network in comparison to the original performance of TCP over the IEEE 802.11 MAC.

### C. Discussion

Alongside our mainstream research on using the ingress throttling to improve the performance of the TCP protocol in multihop ad hoc networks we found two complimentary use cases for our approach. In this section we discuss the application of our way to estimate the capacity of a geographical region for determining scaling limits of ad hoc specific simulations. We devote a separate section below to describe our methodology for the analysis of TCP+routing interactions in ad hoc networks derived from the usage of the ingress throttling technique.

As we have observed in this section, enabling our ingress throttling scheme results in an increased numbers of connections that might be active simultaneously. In [4] it is also demonstrated that fair TCP communications are possible for very large number of connections with long routing paths. However a valid question is how useful this finding is from the end user's point of view? Obviously, for an arbitrary data session there is a limit on the number of competitors in its bottleneck L-region after which the resulting rate throttling limit becomes so small that it would not satisfy even minimal user's expectations. We call this limit the *effective ad hoc horizon*.

1) *Effective ad hoc horizon*: For the particular data session we call a combination of the network state (the number of active sessions in the L-region, the suggested C-load share) and the  $(h, MSS, TX_{802.11})$  parameters of this data session which results in the useful minimal throughput of this data session the "*effective ad hoc horizon*".

In order to illustrate this concept we construct the horizon for a TCP session with  $MSS = 600$  B which might use one of the available 802.11b transmission rates and traverse different number of hops. Firstly, we define the meaningful minimal TCP end-to-end throughput  $Thr_{min} = 30$  kb/s. Secondly, we estimate the maximally achievable throughput of an isolated TCP session for different physical layer transmission rates and number of hops traversed by the flow  $(Thr_{max}(h, MSS, TX_{802.11}))$ . Now dividing the estimated maximal throughput by  $Thr_{min}$  reveals the maximal number of the competing connections inside the L-region for our flow:

$$Num_{max} = \left\lfloor \frac{Thr_{max}(h, MSS, TX_{802.11})}{Thr_{min}} \right\rfloor. \quad (3)$$

Fig. 6 show the number of connections crossing  $h$  hops for each physical layer transmission rate beyond which the end-to-end throughput of a particular flow is below the minimally acceptable threshold. The *effective ad hoc horizon* allows us to draw an important conclusion in the scope of simulation

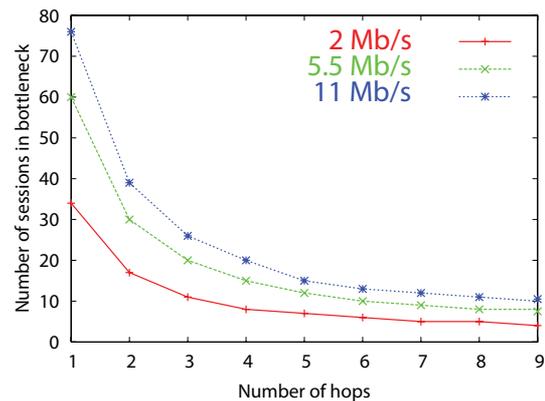


Fig. 6. Effective ad hoc horizon: The maximal number of simultaneously active TCP sessions in the bottleneck L-region.  $MSS=600$ B.

based studies of the IEEE 802.11 based MANETs. It should serve as a reference for the construction of network topologies and the choice of traffic settings in the experiments intended to show scalability of one or another protocol.

Consider the following example. We create a scenario which potentially allow communications over nine hops and formation of the common bottleneck L-region. This corresponds, for example, to a stripe-shaped area with the length and width equal to 1200 m and 100 m correspondingly. Assuming 250 m transmission range (commonly used in simulations), in this area it is possible to form a nine hop flow with internode distance 130 m. Assume we create 20 TCP sessions in this region and at least one session follow a path of nine hops. Assume also that this session uses  $MSS = 600$  B and the physical layer transmission rate is equals 2 Mb/s for all nodes carrying its traffic. The maximally achievable throughput of a single nine hop TCP flow in isolation is 144 kb/s. We can calculate the expected throughput for our nine hop session from (3). The resulting value is only 7.2 kb/s; Obviously this is too low throughput for even a minimal user's expectation. Our concept of the effective ad hoc horizon suggests that in this scenario we should not generate more than *four* TCP sessions to achieve the meaningful expected throughput of 30 kb/s.

## IV. USING INGRESS THROTTLING FOR MEASURING ADMISSIBLE ROUTING LOAD

The vast majority of research on benchmarking different ad hoc routing protocols is mainly conducted using UDP based CBR traffic. It is the unstable and unpredictable performance of TCP in MANETs which motivated the choice of CBR traffic in pioneering (e.g. [12]) and subsequent studies of ad hoc routing protocols<sup>3</sup>.

To a large extend there is no established methodology for the analysis of TCP+routing interactions in MANETs.

<sup>3</sup>Quote from [12] "As the goal of our simulation was to compare the performance of each routing protocol, we chose our traffic sources to be constant bit rate sources. ... We did not use TCP sources because TCP offers a conforming load to the network, meaning that it changes the times at which it sends packets based on its perception of the network's ability to carry packets. As a result, both the time at which each data packet is originated by its sender and the position of the node when sending the packet would differ between the protocols, preventing a direct comparison between them."

Surprisingly, there are very few papers reported on this subject: [13], [14], [15], [16] are the most representative contributions. In this section we show how applying the major properties of our ingress throttling solution we may extract the degree of TCP performance degradation due to routing traffic.

#### A. Considered routing protocols

For the experiments we use stable implementations of AODV-UU and LUNAR which are available from Uppsala University ad hoc implementation portal<sup>4</sup>.

Two variants of AODV protocol are considered. In the first variant the HELLO mechanism is enabled to maintain the connectivity between neighbors further on we refer to this variant of AODV as to AODV-HELLO. In the second AODV variant, further referred as to AODV-LL, the route maintenance is done by means of explicit link layer feedback. In the former case the loss of connectivity between nodes forwarding traffic of a specific connection is detected by three missing HELLO messages; in this case the route maintenance procedure is invoked and the problem is signaled back to corresponding sources. In AODV-LL a packet loss during the transmission is detected by the link layer; the problem is then immediately reported to the routing engine, which in turn invokes the route maintenance operations.

The first variant of LUNAR uses standard settings as described in [17]. The second variant has a modified periodic refresh timer as we describe later on in Section IV-C.6.

Overall, the two variants of AODV and the two variants of LUNAR produce *four* distinct patterns of routing traffic in the networks. These patterns fit into *two* major types of the broadcast traffic invocation: 1) Error-driven (AODV-LL and opportunistic LUNAR) and 2) periodic (AODV-HELLO and LUNAR).

#### B. Current performance metrics and why they are not informative for evaluating TCP+routing interactions

The major problem with the evaluation of TCP+routing interactions is the absence of TCP specific performance metrics for ad hoc routing protocols. In the literature related to the performance analysis of routing schemes we found very few *quantitative* performance metrics. The informational RFC 2051 [18] summarizes these metrics as follows:

- 1) Average number of control routing bits transmitted per data bit delivered, sometimes referred as to “Normalized routing load”. This metric measures the overhead produced by transmission of routing messages.
- 2) Average number of data bits transmitted per data bit delivered, also referred as to “Packet delivery ratio”. This metric is normally interpreted as a measure of the quality of data delivery within the network.

These two metrics are used mainly for two purposes. Firstly, to quantify a load produced by routing traffic and secondly, to assess the effect of routing on UDP based communications for which the packet loss rate is an illustrative performance

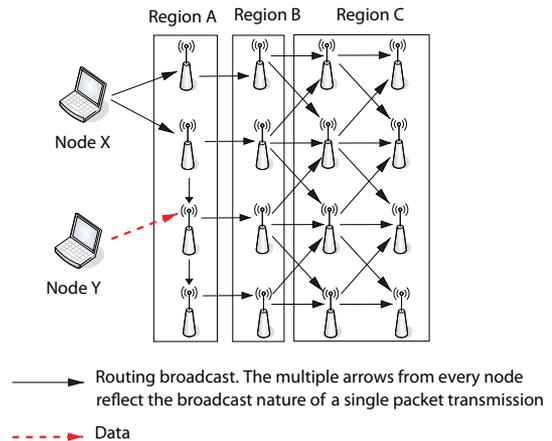


Fig. 7. Illustrative example of inability of the “normalized routing load” metric to capture the spatial effect of routing on data transmission.

characteristic. However, in the case of TCP communications, none of them is able to characterize the overall quality of the ongoing sessions.

As an example of why current performance metrics are not informative consider a scenario depicted in Fig. 7 and assume that the propagation of a routing message issued by Node X causes a loss of one data packet issued by Node Y. During the measurement period we observe 13 transmissions of routing messages in the entire network. The question is: How relevant is the information that 13 routing packets caused the loss of one data packet?

In fact only transmissions in Region B may cause the loss of our data packet if these transmission events coincide in time. If the transmission from Node Y happens at the same time as routing transmissions in Region A, Node Y will refrain from the transmission because it will sense the medium as busy. The nodes in Region C are located three hops away from Node Y, therefore transmissions from this region would not interfere with our data packet. Finally, only transmissions in Region B will cause the loss of the data packet because of the hidden terminal effect. Therefore the actual effect of routing traffic on the data transmission in our example is four routing packets per one lost data packet. The major conclusion from this simple observation is that the “normalized routing load” metric does not reflect the *spatial* effect from the routing activity on the quality of data sessions.

As for the metric “Packet delivery ratio”, it is indeed an informative measure of the routing activity but only with respect to UDP traffic. Evaluation of TCP performance is a more challenging task. The quality of TCP communications is characterized by a *set* of metrics including the throughput and the fairness metrics. In this respect knowing the proportion of the number of delivered TCP data segments to the number of emitted data segments, which would be captured by the “Packet delivery ratio” metric, is certainly not enough to assess the degree of service degradation for the particular TCP flow.

We see the assessment of the routing performance based on direct observation of transmission events for routing packets as problematic due to obvious difficulties of capturing the *spatial* effect of the routing traffic described above. We adopt the

<sup>4</sup>[Online]. Available: <http://core.it.uu.se/AdHoc/ImplementationPortal>

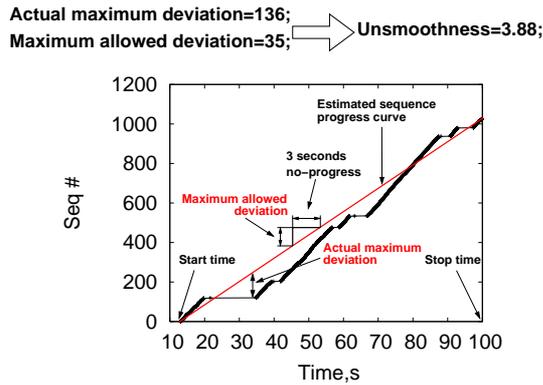


Fig. 8. Computation of the unsmoothness metric

methodology behind the construction of the “Packet delivery ratio” metric and benchmark different routing schemes by monitoring the performance of TCP traffic.

### C. Admissible operation ranges

We use the ingress throttling for achieving stable and predictable performance of the TCP protocol in multihop ad hoc networks with static routing. In our simulation experiments we enable different routing schemes and monitor the deviation of the resulting TCP performance from the reference one.

Up to a certain network scale the effect from transmissions of the control routing messages will be tolerable from the user’s point of view. However, beyond that the routing traffic itself will cause a severe and unacceptable degradation of the TCP quality. We call the set of network configurations (in terms of the network size and the number of sustained TCP sessions) where routing traffic does not become a reason for poor TCP performance the *admissible operation range* (AOR) of MANETs.

Finally, at the border of the AOR, which we call the *routing ad hoc horizon*, we compute the “*equivalent routing load*”. This metric reflects the level of quality degradation for the data traffic caused by a specific ad hoc routing scheme.

Before proceeding further with the description of the AOR experiments we need to introduce new TCP performance metric in addition to those used in Section III.

1) *Definition of the “unsmoothness” metric*: As it is shown in [2] it is important to account for interruptions in TCP flow progress since some of the flows may suffer from long no-progress intervals although the throughput and fairness metrics would report positive network performance. The “unsmoothness” metric captures this qualitative aspect of TCP communications and is constructed as illustrated in Fig. 8. The construction of the metric is done during the analysis of communication traces. After the end of each test run we parse the packet trace and record the progress of the received sequence numbers in time for each flow and the corresponding start times for every session. The stop time for all sessions is assumed to be the same. With this information we estimate the ideal curve of the sequence number progress. It is shown by the straight line in Fig. 8. After that we analyze the recorded process of sequence number arrivals for each flow and compare

each received sequence number with the estimated “smooth” value for the corresponding time. The result of the comparison is the absolute deviation of the actual sequence number from the estimated value. Obviously, in most of the measurements we will have some deviation from the estimated curve even for a perfectly smooth flow due to rounding errors. In order to allow some small deviations we compute the maximum allowed deviation. This value corresponds to the three seconds no-progress time which was chosen based on our empirical observations that longer no-progress times definitely indicate the presence of the TCP unfairness in ad hoc networks. Finally, we compute the “unsmoothness” metric as in (4).

$$Unsmoothness = \frac{\max(|Deviation_{actual}|)}{Deviation_{allowed}^{max}}. \quad (4)$$

The “unsmoothness” metric is always larger than zero. We say that the quality of a TCP flow is acceptable for an end-user if  $Unsmoothness \leq 1$ .

2) *Description of AOR experiments*: In order to determine the “admissible operation range” we performed a series of simulation based experiments on the topology depicted in Fig. 3. In all experiments the width of the forwarding area is fixed and the density of the forwarding nodes ensures *three* hops communications ( $H = 3$ ). This time we vary two parameters in the network: the number of active flows and the node density in the forwarding area. By increasing the number of simultaneously active sessions we increase the intensity of the on-demand routing traffic. By increasing the number of nodes in the forwarding region we increase the duration of the broadcast bursts, since more nodes are involved in the (re-) broadcasting process. In all experiments the *ingress throttling* is enabled. Each experiment is performed for the particular routing protocol with a fixed number of simultaneously active TCP sessions and a variable number of nodes in the forwarding area of the topology.

We start with the minimum number of the forwarding nodes to ensure three hops connectivity for every TCP connection (two nodes). We run simulations with this configuration up to 30 times and measure the worst *unsmoothness* amongst the competing flows and the *unfairness index* in each run. If after 30 runs the average of the worst unsmoothness is less than 1 and the worst unfairness index (2) is less than 0.1 we increase the number of nodes in the forwarding area and repeat the experiment. We continue to increase the number of forwarding nodes until either of the metrics goes beyond the corresponding threshold. At this point we record the last network configuration (the number of TCP sessions and the network size) and the resulting combined (total) TCP throughput. After that we increase the number of simultaneously active TCP connections and repeat the sequence of experiments.

During the experiments with AODV-HELLO the best measured values of the *unsmoothness* metric is between one and two. For this variant of AODV we determine the “*weak AOR*” as it is described below.

The results for the two variants of AODV protocol are shown in Fig. 9 and for LUNAR in Fig. 10. The shaded areas in the figures show the admissible operation range where

the *unsmoothness-unfairness* property is acceptable for all competing TCP flows. Outside the shaded area either the value of the *unsmoothness* or the *unfairness* metric becomes unacceptable for at least one end user.

The left slopes of the AOR show the minimal network configurations where the particular number of distinct connections is possible. For example, four distinct three-hop flows are possible when we have one source node, four destinations and two forwarding nodes, that is in total 7 nodes in the network. The right border of the corresponding AORs represents larger network configurations for the particular number of TCP sessions. For example, in the case where AODV with link layer feedback is used, four connections with the acceptable *unsmoothness* and *unfairness* metrics can exist in a network with four distinct sources, four destinations and 22 forwarding nodes, that is 30 network nodes in total.

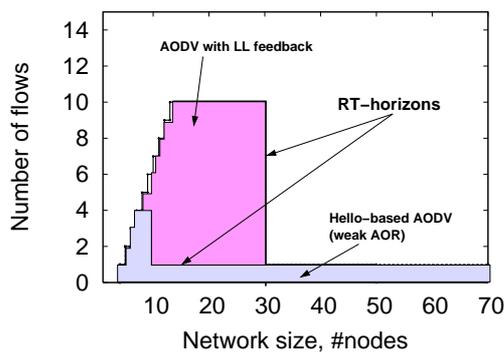


Fig. 9. Admissible operation ranges for AODV-LL and AODV-HELLO.

3) *AODV-LL*: As we observe from Fig. 9 the least impact on the *ideal* TCP behavior is introduced by the routing traffic pattern of AODV with link layer feedback enabled. This is because of the error-driven invocations of broadcast transmissions. In AODV-LL the broadcast activities are initiated as a reaction to packet losses reported by the link layer to the AODV engine. In the case of a packet loss the protocol assumes that the connectivity to the corresponding neighbor is lost and initiates the route recovery procedure. When the ingress throttling scheme is enabled, the collisions between data packets are not frequent. In this case the reason for a higher packet loss rate, hence more frequent invocation of the path recovery phase, is the routing activity itself. As we observe in networks of up to 30 nodes, the broadcast traffic does not introduce enough overhead to force TCP flows into the routing induced slow start phase. However, beyond 30 nodes the broadcast bursts caused by every lost packet are long enough to cause the invocation of the slow start phase at a TCP sender; this leads to stammering TCP flow progress and worse fairness figures.

4) *AODV-HELLO*: The most unstable effect on the quality of TCP sessions is introduced by the traffic patterns produced by AODV-HELLO. In all experiments the *unsmoothness* of the competing TCP flows was between one and two at best. The analysis of the communication traces reveals that this is due to transmissions of HELLO messages. Despite of the *small* size, their frequent and independent emissions from multiple

nodes does not allow any of the competing TCP sessions to progress smoothly through the network. Even though we could not observe perfectly smooth flows as we did in the case of AODV-LL, we decided to delimit the admissible operation range where  $0 < Unsmoothness < 2$  and the unfairness index is less than 0.1. We labeled this admissible operation range for AODV-HELLO as “weak AOR” in Fig. 9.

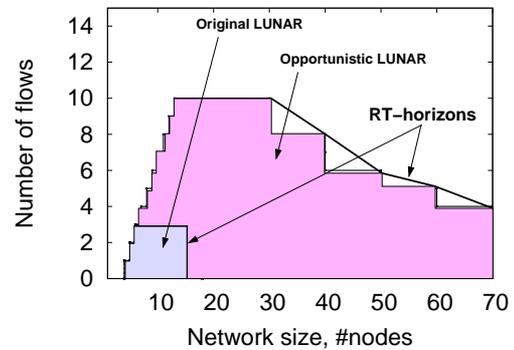


Fig. 10. Admissible operation ranges for original and opportunistic LUNAR.

5) *LUNAR*: As for the effect of the routing traffic pattern generated by LUNAR, we observe that it is very similar to that of AODV-HELLO. The smaller shaded area in Fig. 10 shows the admissible operation range for the original LUNAR with its forced three seconds complete route re-discoveries. The difference is that the horizon with LUNAR is stable, meaning that the *unsmoothness* metric for all flows within the AOR is equal to or less than 1 and the fairness is close to perfect. The stability of LUNAR’s admissible operation range can be explained by less frequent initiations of the broadcast traffic than in the case of AODV-HELLO. However, in the case of LUNAR we have a strict right border. This can be explained by the difference in the nature of broadcast patterns generated by the two protocols. If in the case of AODV-HELLO we have short frequent one hop broadcasts, in the case of LUNAR we have less frequent but more massive flooding waves, which result in stammering TCP progress with increasing number of nodes in the network.

6) *Opportunistic LUNAR*: As can be seen in Fig. 10 the original LUNAR has a narrow admissible operation range. As an experiment we changed the route refresh strategy of LUNAR to one which closely resembles the error-driven pattern of AODV-LL. To do this we disabled the forced three second complete route rediscovery mechanism at sources. Instead, we retain a route as long as there are packets arriving to the forwarding engine: With every new data packet we shift the route timeout three seconds into the future. This modification allowed us to create an “*opportunistic*” version of LUNAR.

This change allowed us to significantly extend the horizon. Moreover, we achieved even better characteristics in comparison to AODV-LL. Now the right edge of the admissible operation range stretches to bigger networks. This is because we do not interpret every loss of a packet as an indication of the route breakage as it is the case with AODV-LL. Instead, we react on invocations of slow starts in TCP which are less

frequent events, given that the ingress throttling scheme is deployed in all sources. By doing so we significantly decrease the frequency of broadcast bursts.

#### D. Routing ad hoc horizon and equivalent routing load

Above we presented the reasoning for why the estimation of the routing load is complex and not an obvious task. Recapitulating, this is due to the difficulties to capture the spatial effect of the broadcast traffic on a particular data session. Because of this the averaged metrics for routing load do not reflect the spatially distinct effects of the routing traffic on the ongoing data sessions. In the previous subsection we described a way to quantify the admissible operation region for reactive ad hoc routing protocols and TCP communications. The *routing ad hoc horizon* represents the critical network size where TCP flows maximally use the network capacity which is not consumed by transmission of routing messages and have an acceptable end user quality. Beyond this horizon, TCP flows begin to suffer from routing induced unfairness. In this case the progress of affected flows becomes disrupted, potentially leading to long disconnection intervals and loss of network utilization. The important property of the stability of communications within the AOR is that we can indirectly estimate the routing load produced by one or another protocol.

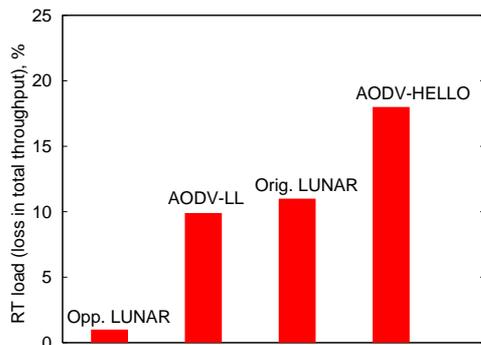


Fig. 11. Equivalent routing load for the considered routing protocols.

We define an “*equivalent routing load*” metric as follows. Assuming that our ingress throttling is deployed in all sources and that we measure the total TCP throughput in the network without routing ( $Thr_{norouting}^{tot}$ ), in the case where all competing flows are active the network is fully utilized. Now we measure the total TCP throughput on the right border of the admissible operation range ( $Thr_{withrouting}^{tot}$ ). In this case the competing TCP flows maximally use the capacity that is left from the routing traffic. Then the quantity  $(1 - \frac{Thr_{withrouting}^{tot}}{Thr_{norouting}^{tot}}) \cdot 100\%$  reveals the reduction in the throughput in per-cent<sup>5</sup> due to the routing activity. Fig. 11 shows the equivalent routing load for the considered protocols.

We may observe that in addition to the very narrow admissible operation range the HELLO-based AODV consumes

more useful data throughput than other protocols. As for the modified opportunistic LUNAR protocol, its “equivalent routing load” is the smallest among all routing protocols. This is because the routing activity in this case happens only during the path establishment of each flow. After that the ingress throttling scheme prevents invocation of slow starts in the stable network as we have in our experiments. Therefore no further routing traffic is involved after all sessions were successfully established until the end of simulations.

#### E. Summary

In this section we developed our approach for the analysis of the effect of ad hoc routing traffic on the quality of TCP communications. For each of the the two considered routing protocols and static topologies we determined the admissible operation range, which is the set of network configurations where routing and TCP traffic can peacefully co-exist.

We found that 10 - 20 nodes is a critical network size for routing schemes which utilize periodic broadcast invocations in their route maintenance phase. With such protocols only few TCP connections may exist with an acceptable quality. It is worth noting that this type of broadcast invocations is present in the specification of a new generation reactive routing protocol for MANETS. In the recent draft for *DyMo* [19] the HELLO mechanism for the connectivity maintenance between the neighbors is inherited from AODV. In contrast, our analysis shows that a “TCP-friendly” ad hoc routing protocols should use mainly an error-driven form of broadcast invocations.

## V. CONCLUSIONS

In this article we considered the problem of the severe unfairness between multiple multihop TCP flows in a wireless ad hoc network. We described the ingress throttling scheme which enforces the wireless specific fairness model and dramatically extends the operation range of ad hoc networks.

We also analyzed the effect of different routing traffic patterns on the performance of TCP during stable operations of MANETS. We highlighted the fact that the current performance metrics for the evaluation of routing protocols are not informative enough for the evaluation of TCP+routing interactions. Based on the properties of the ingress throttling scheme we suggested a methodology for an indirect measurement of the routing load and quantification of the routing effect on TCP performance. Using this technique we were able to show that the routing traffic itself can be a reason for poor TCP performance in MANETS. Our major conclusion regarding the effect of routing traffic patterns on TCP communications is that periodic, non error-driven broadcasts of even short messages is harmful for data communications and leads to narrowing the operational region of MANETS.

Further studies have to add mobility to the scene, which is another limiting factor for the unstable performance of transport protocols in multihop ad hoc networks.

## REFERENCES

- <sup>5</sup>We intentionally do not express the “equivalent routing load” in bits per second since this value would be specific to the particular transmission speed picked in the experimental environment (in our case ns-2).
- [1] K. Xu, S. Bae, S. Lee, and M. Gerla, “TCP behavior across multihop wireless networks and the wired Internet,” in *Proc. WoWMoM’02*, Atlanta, GA, USA, Sep. 2002.

- [2] Ch.Tschudin and E. Osipov, "Estimating the ad hoc horizon for TCP over IEEE 802.11 networks," in *Proc.MedHoc'04*, Bodrum, Turkey, Jun. 2004.
- [3] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine*, Jun. 2001.
- [4] E. Osipov and Ch.Tschudin, "TCP-friendly bandwidth sharing in mobile ad hoc networks: From theory to reality," *EURASIP Journal on Wireless Communications and Networking. Special issue "Mobile Multi-Hop Ad Hoc Networks: from theory to reality"*, 2007.
- [5] —, "A path density protocol for MANETs," *Ad Hoc and Sensor Wireless Networks (Old City Publishing)*, 2007.
- [6] G. Xylomenos, F. C. Polyzos, P. Mähönen, and M. Saaranen, *TCP performance Issues over Wireless Links*, M. Hassan and R. Jain, Eds. Prentice-Hall, 2004.
- [7] A. A. Hanbali, E. Altman, and P. Nain, "A survey of TCP over mobile ad hoc networks," INRIA, France, Tech. Rep. 5182, May 2004.
- [8] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED," in *Proc. ACM MobiHoc'03*, Annapolis, MD, USA, 2003.
- [9] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE JSAC*, vol. 19, no. 7, 2001.
- [10] R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmissions," *ACM Comp. Commun. Rev.*, vol. 30, no. 1, Jan. 2000.
- [11] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE Infocom'03*, San Francisco, CA, USA, 2003.
- [12] J. Broch, D. Maltz, D. Johnson, Y-C.Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM MobiCom'98*, Dallas, TX, USA, 1998.
- [13] T. Dyer and R. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," in *Proc. ACM MobiHoc'01*, Long Beach, CA, USA, Oct. 2001.
- [14] D. Perkins and H. Hughes, "Investigating the performance of TCP in mobile ad hoc networks," *International Journal of Computer Communications*, vol. 25, no. 11-12, 2002.
- [15] K. Nahm, A. Helmy, and C.-C. J. Kuo, "TCP over multihop 802.11 networks: issues and performance enhancement," in *Proc. ACM MobiHoc'05*, Urbana-Campaign, Illinois, USA, May 2005.
- [16] C. Rohner, E. Nordström, P. Gunningberg, and C. Tschudin, "Interactions between TCP, UDP and routing protocols in wireless multi-hop ad hoc networks," in *Proc. IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN'05)*, Santorini, Greece, Jul. 2005.
- [17] C. Tschudin, R. Gold, O. Rensfelt, and O. Wibling, "LUNAR: a lightweight underlay network ad-hoc routing protocol and implementation," in *Proc. NEW2AN'04*, St. Petersburg, Russia, Feb. 2004.
- [18] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," IETF Informational RFC 2501, 1999. [Online]. Available: <http://www.rfc-editor.org/rfcsearch.html>
- [19] I. Chakeres, E. Belding-Royer, and C. Perkins, "Dynamic MANET on-demand (DYMO) routing," IETF draft (work in progress), 2005. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-02.txt>



**Evgeny Osipov** received the Engineer degree from Krasnoyarsk State Technical University, Russia in 1998, MSc degree from EPFL, Swiss Federal Institute of Technology in 1999, Licentiate of Engineering from KTH, Royal Institute of Technology, Sweden in 2003 and PhD in Computer Science from University of Basel, Switzerland in 2005. Presently he is a postdoctoral fellow at RWTH, Aachen University, Germany. His major research interests are in architectures and protocol design for wireless sensor and computer networks.



**Christian Tschudin** is a full professor for computer networks at the University of Basel, Switzerland. Before joining the University of Basel, he was at Uppsala University as well as ICSI in Berkeley and did his PhD in Computer Science at the University of Geneva. He has a masters degree in Mathematics. Christian Tschudin's research interests are in mobile code, wireless networks, network architecture and security.