

Research Article

TCP-Friendly Bandwidth Sharing in Mobile Ad Hoc Networks: From Theory to Reality

Evgeny Osipov¹ and Christian Tschudin²

¹Department of Wireless Networks, RWTH Aachen University, 52072 Aachen, Germany

²Computer Science Department, University of Basel, 4056 Basel, Switzerland

Received 30 June 2006; Revised 13 December 2006; Accepted 11 January 2007

Recommended by Marco Conti

This article addresses a problem of the severe unfairness between multiple TCP sessions in a wireless context also known as “TCP capture” phenomenon. We present an adapted max-min fairness framework to the specifics of MANETs. We introduce a practically implementable cross-layer network architecture which enforces our formal model. We have verified with simulations and real world experiments that under our adaptive rate limiting scheme unfair behavior virtually vanishes. The direct consequence of this work guarantee stable services for TCP-based applications in MANETs, including traditional FTP, web as well as for UDP-based sessions.

Copyright © 2007 E. Osipov and C. Tschudin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Poor and unstable TCP performance over multihop wireless links is one of the stumbling blocks which prevents mobile ad hoc networks (MANETs) from wide deployment and popularization. TCP capture, as is described in [1], is one of the unsolved problems in MANETs which manifests itself in extremely unfair distribution of network bandwidth between competing sessions.

The problem of the unfairness is a feature of inadequate behavior of the congestion control mechanism of TCP in radio transmission medium. The major assumption for TCP's congestion control mechanism is that missing acknowledgments for data segments are signals of network congestion. However, this assumption does not hold in wireless environment where a high bit-error rate, unstable channel characteristics, and user mobility largely contribute to packet corruptions. As a result of the erroneous interpretation of radio collision induced packet losses as a network congestion, TCP reduces its rate and its throughput decreases. This phenomenon was initially observed in single hop wireless networks [2, 3].

The situation further exacerbates in the multihop case. In MANETs we have a “super-shared” medium where multihop links belong to the same radio collision domain. The problem here is a presence of a so-called interference zone of ra-

dio transmitters. This is a geographical area where the radio signal cannot be correctly decoded by the receiving station, however its power is high enough to cause losses of packets transmitted by nodes in the assured radio reception range. As it is experimentally shown in [4] 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. This leads to the formation of a narrow “ad hoc horizon” located at two to three active TCP connections each following a path of up to three hops. Beyond this scale the quality of communications becomes unacceptable for an ordinary end-user.

This article describes a cross-layer network architecture for MANETs that does not require overloading standard MAC and TCP protocols with wireless-specific fixes. The formal part of the architecture is an adapted max-min fairness model to the specifics of multihop wireless communications and an adaptive distributed capacity allocation algorithm. The practical part suggests an implementation of the *ingress rate throttling* scheme that enforces the model and solves the unfairness problem. The major improvements we achieved by throttling the output rate at the ingress nodes are an increase in total network throughput and almost perfect fairness. These properties we demonstrate by simulations and real-world experiments.

The problem of fairness in a multihop wireless context has been extensively studied during the last several years. However, few works suggest both theoretically supported and practically implementable solutions. Our reflection of the traditional max-min fairness model for the case of MANETs has some similarities with the approach suggested in [5, 6]. However, a different interpretation of the formal model parameters and an original implementation strategy make our work distinct from the above-mentioned. We comment on the major differences in a separate section below.

The rest of the article is organized as follows. In Section 2 we outline the major design options for the ingress throttling scheme and the overall architecture. We present our interpretation of the max-min fairness model and the mechanism for its enforcement in MANETs in Sections 3 and 4, respectively. After that in Section 5, we report on performance results of our solution obtained using simulations and real-world measurements. We discuss the related work and open issues in Section 6 before concluding with Section 7.

2. SOLUTION OUTLINE

In this article we consider only connected networks where there is a potential multihop path between any pair of nodes. By assuming this we also limit the scope of considered ad hoc networks to medium size, community-based formations. We see this type of networks as one the most probable applications of ad hoc networking. The solutions aiming at minimizing the effect of mutual cross-cluster radio interferences in disconnected networks include adding heuristics to the congestion control of the TCP protocol (e.g., [7]) various smart channel management techniques (e.g., [8]) and the usage of directional antennae. While these kinds of solutions show promising results, they either require serious modifications to the existing and widely deployed hard- and software or simply the technology is not yet available for an ordinary user. On contrary, our primary goal is to achieve stable operation of traditional Internet services in ad hoc networks built on currently available and widely spread IEEE 802.11 technology. By this we intend to bridge the gap between the research promises of “ad hoc” benefits and the reality where the ad hoc networking to a large extent does not exist.

We primarily concentrate on achieving fairness for TCP-based communications. During the course of our work, we understand that this approach gives us necessary insights for achieving fairness in networks with heterogeneous data traffic. We describe the fundamentals of our approach assuming the case of static routing and no mobility. However, this assumption is relaxed on the stage of implementation which is reflected in the experimental assessment part of this article. Otherwise, we do not place additional assumptions: the competing data flows may use any available transmission rates of IEEE 802.11b at the physical layer, traverse different numbers of hops, and use variable packet sizes.

At first we formally introduce a fairness framework for MANETs. For this we adapted the fairness model from the wireline Internet to the specifics of the multihop wireless environment. The major outcome of this stage is new def-

initions of *bottleneck regions* and the *boundary load* within them as analogs to the wireline bottleneck link and the capacity terms. With these newly defined terms we shift the focus from the *link-capacity* domain, specific to the wireline networks, to the MANET specific *space-load* domain. We propose an algorithm for load distribution between the connections competing inside the bottleneck regions.

On the second stage 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. In its simplest form, the rate limit is a function of

- (i) the number of hops for a particular connection;
- (ii) the underlying physical-layer transmission rates along a path of the particular connection;
- (iii) the number of competing connections on the path of the considered connection (path density).

These parameters are feasible to obtain using the facilities of ad hoc routing protocols as described in [9]. We apply the derived rate limit to configure a scheduler at the interface queue of sources of competing sessions (the *ingress nodes*) and shape the outgoing traffic accordingly. This implementation decision eliminates the need for overloading standard MAC and TCP protocols with MANET-specific fixes. With this scheme none of the TCP sessions is able to benefit from temporal weaknesses of the competitors by capturing the transmission capacity.

3. MAX-MIN FAIRNESS IN SPACE-LOAD DOMAIN

Before proceeding further with the definition of a framework for fairness in MANETs, let us recall the traditional network model and the definition of fairness used in the wireline context.

D1 Network model

Consider a set of sources $s = 1, \dots, S$ and links $l = 1, \dots, L$. Let $\Theta_{l,s}$ be the fraction of traffic of source s which traverses link l and let C_l be the capacity of link l . A feasible allocation of rates $r_s \geq 0$ is defined by $\sum_{s=1}^S \Theta_{l,s} r_s \leq C_l$ for all l .

D2 Bottleneck link

Based on the network model defined above, link l is said to be a bottleneck for source s if and only if

- (1) link l is saturated: $C_l = \sum_i \Theta_{l,i} r_i$;
- (2) source s on link l has the maximum rate among all sources using link l : $r_s \geq r_{s'}$ for all s' such that $\Theta_{l,s'} > 0$.

D3 Max-min fairness

A feasible allocation of rates \vec{r} is “max-min fair” if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller rate. This happens when every source has a bottleneck link.

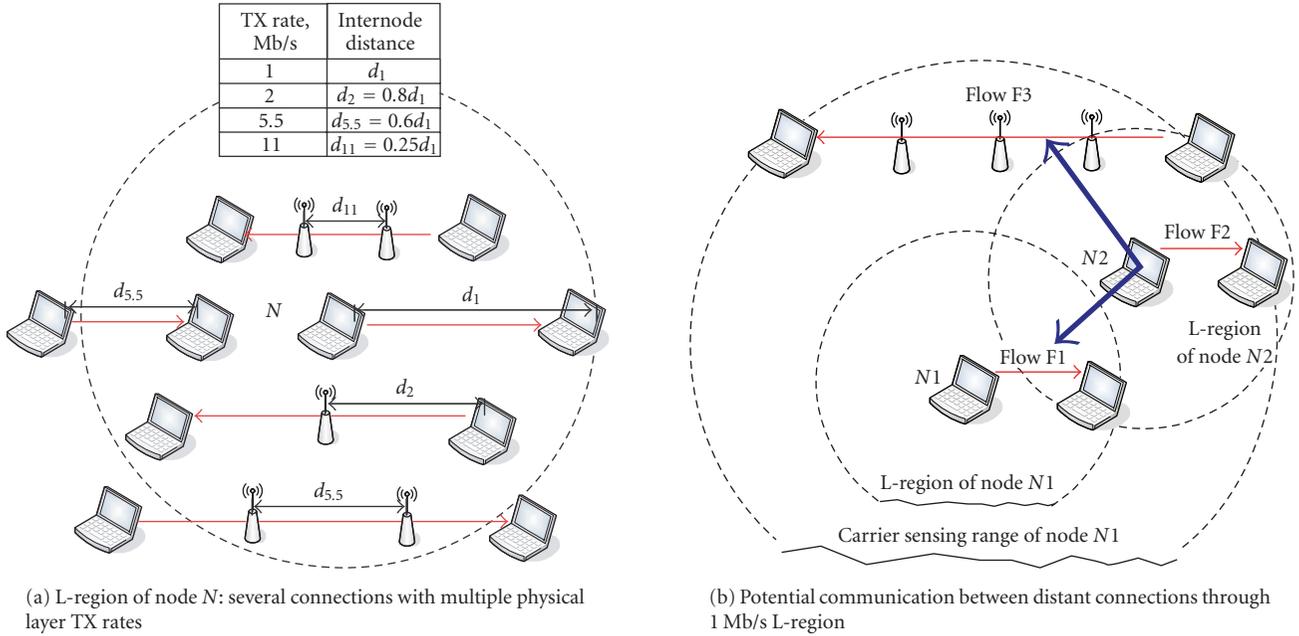


FIGURE 1: Illustration of the L-region concept.

3.1. Reflecting the model parameters to the case of MANETs

Apparently the major stumbling block in reflecting the above network model and defining the fairness criteria in the case of MANETs is a notion of the *link* and the associated terms *capacity* and *rates of sources* on the link. Below we present definitions of functional analogies of these terms in the multihop wireless domain.

From wireline “link” to wireless “L-region”

In general for IEEE 802.11-based networks the term “link” is misleading. Obviously, it is incorrect to consider an imaginary line between two communicating nodes as the link since the radio signal from a given packet transmission is propagated in a geographical region of a certain size.

We define the *L-region* as an area around a wireless node equal to the size of the 1 Mb/s transmission range of an IEEE 802.11 radio transmitter traversed by at least one end-to-end data flow.

The concept of the L-region is illustrated in Figure 1(a), where d_1 is the internode distance that equals the radius of the 1 Mb/s transmission zone.¹ The scale of the figure is chosen according to the results of the real-world measurements of communication ranges for different transmission rates of IEEE 802.11b devices in [10]. Note that in reality the shape of L-region is complex and is not an ideal circle as the fig-

ure shows. However, defining the L-region as the 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.

The rationale for defining L-region as 1 Mb/s transmission range is virtually the same as behind the virtual carrier sensing with RTS/CTS. We need means of communication between nodes carrying traffic of competing connections. With such definition two connections 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 Figure 1(b) where Flow F1 and Flow F3 can discover the presence of each other through L-region of node N2. Note that in the figure node N2 carries data traffic of Flow F2, however this is not a requirement. In general, the central node of an L-region itself may or may not carry data of an end-to-end data flow; its presence assures potential communication between distant connections by means of network protocols.

From wireline “sources” to wireless “associations”

Defining the L-region as a geographical region with the *functional* properties of the wireline link we need to reconsider the concept of the data “source.” On a wireline link a part of capacity is consumed by packet transmissions from a single entity—the session’s source. As it is illustrated in Figure 1 the L-region is sparse enough to accommodate different numbers of nodes transmitting packets of a specific end-to-end data flow depending on the used transmission rate at the physical layer. Obviously, transmissions from all nodes that carry traffic of a specific connection located inside the L-region consume its capacity.

¹ For the sake of clarity in this and the following 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.

We define the *source-destination association* as a set of nodes including the source, destination, and the nodes forwarding packets of a specific data flow. We say that a node of a particular 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.

From wireline “rate” and “capacity” to wireless “C-load share” and “boundary C-load”

The notion of “rate” in wireline networks relates to the notion of “capacity” of the link. On a given link the rate of traffic from a particular source is a fraction of the link capacity $r_s \leq C_i$. Thus the term “rate” makes sense only when the term “capacity” is well defined and its value is finite. In the case of the L-region the later term is impossible to identify uniquely in conventional bits per second. This is because in general nodes inside the L-region may use any of the available physical-layer transmission rates.

As a resource to share within the L-region, we define a *load* which competing associations *generate* or *consume* inside the L-region. We refer this term as to *conserved load* (C-load) and normalize the *boundary C-load* to one.

We define *C-load share* (ϕ) to be an analog to the wireline “rate”: it is a fraction of the *boundary C-load* that the particular connection generates or consumes inside the L-region.

3.2. Max-min fairness in space-load domain

With the above-defined MANET-specific substitutes for the *source*, the *link*, the *rate* of sources, and the *capacity* of the links, we formulate the *space-load* max-min fairness as follows.

D3 MANET network model in the space-load domain

We consider a set of associations $a = 1, \dots, A$ and the set of existing L-regions $\lambda = 1, \dots, \Lambda$. Let $\Gamma_{\lambda,a}$ be an indicator of the presence of association a inside L-region λ :

$$\Gamma_{\lambda,a} = \begin{cases} 1, & a \in \lambda, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A feasible allocation of C-load shares $\phi_a > 0$ is defined by $\sum_{a=1}^A \Gamma_{\lambda,a} \phi_a \leq 1$ for all L-regions λ .

D4 Bottleneck L-region

With the space-load MANET model defined above, we define a bottleneck L-region for the association a if and only if

- (1) L-region λ is saturated: $\sum_i \Gamma_{\lambda,i} \phi_i = 1$;
- (2) association a in L-region λ has the maximum C-load share among all associations located in L-region λ : $\phi_a \geq \phi_{a'}$ for all a' such that $\Gamma_{\lambda,a'} = 1$.

D5 Max-min fairness in space-load domain

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 belongs to a bottleneck *L-region*.

The proof of the above fairness criterion resembles the proof of a similar theorem in [11] and is omitted for the reason of limited space.

3.3. The algorithm of C-load shares distribution

For the complete picture of the fairness framework in MANETs we need to describe an algorithm for max-min fair distribution of C-load shares between the associations inside L-regions and suggest a mechanism by means of which the associations conform to the assigned fair shares. In this section we describe a centralized algorithm for C-load share distribution. Our goal is to show feasibility of max-min fair C-load shares distribution in a finite time. In order to simplify the description we assume the following:

- (1) during the execution of the algorithm the network and the set of associations are stable. This means that associations neither leave the initial L-region nor appear in the new L-region. This assumption is relaxed in the distributed implementation of the algorithm, which accounts for sessions mobility;
- (2) initially all associations are not assigned the C-load shares. Further on we refer an association without the assigned load share as to the *fresh association* and an association with the assigned load share as to the *assigned association*;
- (3) all nodes in the network execute the same algorithm and cooperate;
- (4) the information is distributed between the MANET nodes by means of a message passing scheme. However for a general description of the algorithm we do not suggest any particular protocol and assume that all necessary information is accessible at a centralized control point.

Denote the number of associations inside L-region λ (*L-region density*) as $\rho_\lambda = \rho_\lambda^{\text{fresh}} + \rho_\lambda^{\text{assigned}}$, where $\rho_\lambda^{\text{fresh}}$ and $\rho_\lambda^{\text{assigned}}$ are correspondingly the numbers of fresh and assigned associations inside L-region λ . The algorithm of max-min fair assignment of C-load shares is as follows.

- (1) All central nodes of L-regions suggest a C-load share to the visible fresh associations according to the following formula:
 - (a) L-regions with only fresh associations: $\phi = 1/\rho_\lambda$;
 - (b) L-regions with assigned and fresh flows: $\phi = (1 - \sum_{i=1}^{\rho_\lambda^{\text{assigned}}} \phi_i) / \rho_\lambda^{\text{fresh}}$.
- (2) Among all L-regions choose those which suggest the minimal C-load share. Assign the computed share to

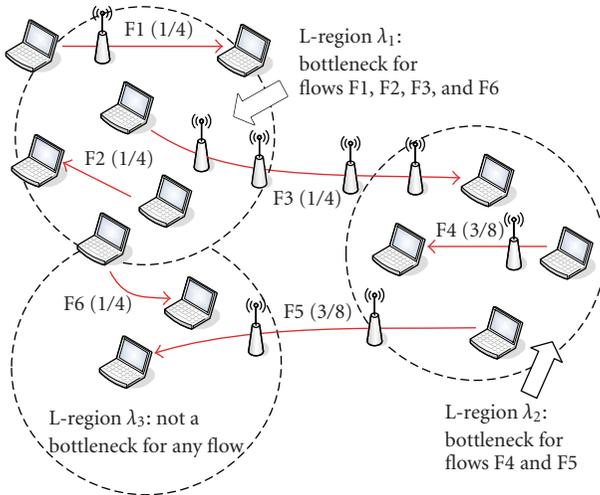


FIGURE 2: The output of the C-load share assignment algorithm.

the associations which are included in these regions. Do not modify the shares of these associations after that.

- (3) Repeat steps (1) and (2) until all flows are assigned the C-load share (all L-regions contain only assigned associations).

The presented-above algorithm terminates because the set of associations and subsequently the set of L-regions are finite. Figure 2 shows the resulting C-load share assignment for a sample network with six end-to-end data flows. With the shown network settings the algorithm terminates after two iterations and detects bottleneck L-regions as is illustrated in the figure. The proof of the max-min fair allocation property of the algorithm is presented in [12].

In [9] we describe a real-world implementation of a distributed version of the algorithm. An extension to a reactive ad hoc routing protocol called the *path density protocol* (PDP) allows delivering the fair C-load shares to the sources of competing connections. PDP utilizes the fact that in reactive routing protocols for IEEE 802.11-based networks route request messages are propagated with the base transmission rate 1 Mb/s. In PDP all network nodes overhear route setup messages and maintain a soft state of end-to-end data flows existing in each one-hop neighborhood. By piggybacking the local state information in each rebroadcasted message every node maintains a consistent view of the competing flows inside L-regions. We do not further discuss the details of the path density protocol in this article and refer the reader to [9].

3.4. Summary of the space-load fairness framework

In this section we reviewed the fairness framework in the wireline Internet. Specifically, we focused on the fairness of sharing the capacity of the links in the case of multihop communications. We recapitulated the network model which is

used to define objectives for max-min fairness in the wireline networks.

We showed 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 MANETs. This is due to specifics of the radio transmission medium. We presented new entities called the *association*, the *L-region*, the *C-load share* and the *boundary C-load*, which serve as substitutes for the corresponding terms in multihop wireline networks.

These newly defined terms allowed us to formulate the max-min fairness criterion for wireless ad hoc networks in the *space-load* domain. Finally, we described a generic algorithm for max-min fair assignment of C-load shares for the competing associations in their bottleneck L-regions.

4. ENFORCEMENT OF FAIR C-LOAD SHARES IN MANETS

Having defined C-load as a unit-less measure for the resource to share in a geographical region we need to give its interpretation in the terms meaningful to the network nodes. In this section we present the mapping of the space-load model parameters back into the rate-capacity domain.

4.1. TCP throughput as a reference to the boundary C-load

The interpretation of the boundary C-load by sources of TCP connections in terms of the transmission rate is somewhat straight forward. We need to find a condition under which every node of an association tends to generate maximal load inside a geographical region. If for a moment we consider the wireline Internet, this condition has a direct analogy in terms of the *bandwidth-delay product*—the amount of traffic that the entire path can accommodate.² For the estimation of the bandwidth-delay product the major property of TCP protocol is used: a single TCP flow in a steady state is a perfect estimator of the available bandwidth in the network. We will use this property for the estimation of the boundary C-load.

Indeed, running along over a multihop MANET a single TCP flow will generate maximal load. In the steady state every node of the particular association has a continuous backlog of packets. If we consider an arbitrary multihop association and potential L-regions with the centers located in the nodes of this association we can always identify the L-region where the TCP connection will constantly be active. Figure 3 shows a constant “air presence” of a single TCP session inside the L-region. In the figure we have a single three hops TCP session from node $N1$ to node $N4$. In the steady state the amount of data packets backlog at node $N1$ will be constant because of continuous arrival of acknowledgments. As it is visible from the figure in the potential L-region with the center in node $N2$, our TCP session constantly produces the load from nodes $N1$, $N2$, and $N3$.

² In the Internet the bandwidth-delay product is used to dimension the *congestion window* parameter at sources to prevent local congestion.

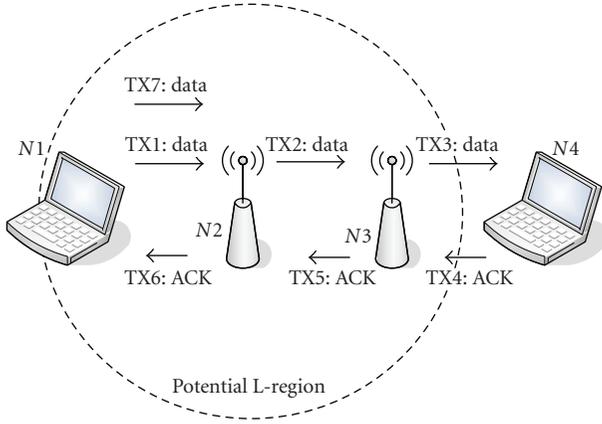


FIGURE 3: Constant presence of a single flow inside an L-region.

$\text{Thr}_{\max}(h, \text{MSS}, \text{TX}_{802.11})$ denotes the maximal throughput achieved by a TCP connection in a network free from the competing data sessions. There h is the number of hops traversed by the flow with $\text{TX}_{802.11}$ transmission rate at the physical layer between the hops. The source generates data segments of size MSS bits. Consider now a network with several competing TCP connections. Each source of TCP sessions interprets this value as an individual reference to the C-load inside its bottleneck L-region. Obviously, the value Thr_{\max} can be different for each connection depending on its characteristics. However, this is exactly the property of the parameter that we need: being unique for the competing connections these values refer to a single common entity—the C-load in the bottleneck L-region.

Practically the above interpretation strategy can be implemented in two ways. The value Thr_{\max} can either be formally estimated or experimentally measured for all combinations of input parameters. For the initial prototyping and experimental performance assessment we chose the second option. We describe some practical issues in Section 4.3. As for the first option, the formal estimation of the maximal TCP throughput in multihop wireless networks is a complex task with no available ready to use solutions. We further comment on this issue in Section 4.1.

4.2. The ingress throttling formula

Taking the maximally achievable throughput by session i as a reference to the boundary C-load in order to conform traffic of this data session to the assigned fair share of the C-load ($\phi_{\text{bottleneck}}$) the output rate from the sources of TCP connection i should be reduced according to the following formula:

$$r_i^{\text{TCP}} \leq \text{Thr}_{\max}(h, \text{MSS}, \text{TX}_{802.11}) \cdot \phi_i^{\text{bottleneck}}. \quad (2)$$

In the above formula, Thr_{\max} is the session's reference to C-load in its bottleneck L-region. The parameter $\phi_i^{\text{bottleneck}}$ is the fair C-load share of TCP session i in its bottleneck assigned by the C-load distribution algorithm described above.

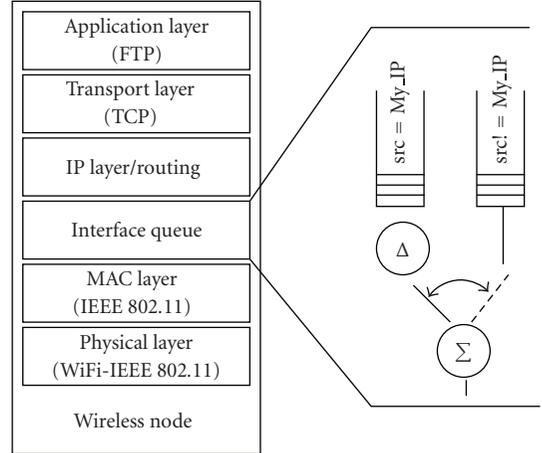


FIGURE 4: Structure of a IEEE 802.11 enabled node.

4.2.1. Treating UDP traffic inside the space-load fairness framework

In order to conform UDP traffic to the space-load fairness framework, the ingress throttling formula (2) should be adjusted to account for the one-way nature of UDP communications. What we should do is to increase the derived rate limit by the fraction corresponding to the transmission of TCP acknowledgments. We do not discuss further performance issues of UDP traffic to the space-load framework in this article and refer to [12] for more details including the experimental performance assessment.

4.3. Rate throttling enforcement at the ingress nodes

Schematically the architecture of a wireless node is presented in Figure 4. For simplicity of presentation we assume that one source originates only one end-to-end connection. We also assume that the interface queue is either logically or physically divided into two queues: one for data packets originated at this node and the second one for all other data packets. These two queues are drop-tail in nature. Upon arrival from the routing layer, all packets are classified according to their source IP addresses and placed into the corresponding queue.

The scheduler from the interface queue to the MAC layer consists of two stages. At the first stage we have a fixed delay non-work-conserving scheduler with a tunable delay parameter Δ . When $\Delta = 0$ the first stage scheduler works as usual work-conserving scheduler and the whole scheduling system works as a scheduler with three priorities. Scheduler Σ at the second stage is a nonpreemptive work-conserving scheduler with the highest priority to the queue with locally generated packets and then to the forwarding queue. The transmission rate limit (2) is used to set parameter Δ (3) of the scheduler for the queue with locally generated packets. We compute the delay parameter as

$$\Delta = \frac{\text{MSS}_i}{r_i^{\text{TCP}}(h, \text{MSS}, \text{TX}_{802.11})}, \quad (3)$$

where MSS_i is the size of the maximum data segment size used by TCP session i .

For the implementation of the above architecture, we experimentally measure the throughput of a single TCP flow with various characteristics in isolation from the competing traffic. After the measurements we construct a table of Thr_{\max} values and make it available to each node. Then the scheduler's processing block at nodes generating the traffic chooses the appropriate value depending on the parameters of the particular session and configures the delay parameter Δ .

4.4. Transmission overhead due to distributed gathering of throttling parameters

Assume that $\text{Thr}_{\max}(h, \text{MSS}, \text{TX}_{802.11})$ —the estimates for the maximal achievable throughput—are hardwired at source nodes and available on-demand. Now at the particular ingress node in order to compute the throttling limit 1, we need (2) to choose the correct value of the maximal throughput according to the characteristics of the particular end-to-end connection and (3) to obtain the fair C-load share for this connection in the network.

Note that out of the three input parameters for $\text{Thr}_{\max}(h, \text{MSS}, \text{TX}_{802.11})$ the maximum segment size (MSS) can be detected locally for every incoming to the scheduler packet. The value for the path length (h) can be easily obtained from route reply messages. The recent results from [13, 14] indicate the feasibility of extending ad hoc routing protocols to find the available IEEE 802.11 transmission rates on the path ($\text{TX}_{802.11}$) with little overhead. As for the fair C-load share parameter, in [9] we suggested the *path density protocol* an extension to a reactive routing scheme that delivers parameter $\phi^{\text{bottleneck}}$ to corresponding TCP sources. The overhead caused by the exchange of information about the presence of competing associations inside L-regions is on average 1 kb/s per connection. By this, our solution satisfies the practical energy and bandwidth efficiency conditions for realistic ad hoc networks.

4.5. Summary of the fairness enforcement strategy in MANETs

In this section we discussed 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.

We suggested to implement the rate limitation at the interface queue of the nodes that generate own traffic. Thus the nodes only forwarding the traffic of other flows do not perform any shaping actions. Overall, our solution does not require changes to the standard TCP nor IEEE 802.11 and is implementable by enhancing routing protocols and the use of traffic policing at the ingress nodes.

5. EXPERIMENTAL ASSESSMENT OF OUR SOLUTION

In the experiments presented below we apply the space-load fairness model to the IEEE 802.11b-based networks. In order to show the level of qualitative performance improvements achieved with the ingress throttling scheme, we present a selected number of experiments with limited number of variable parameters. In particular we present the performance evaluation with TCP flows only. For an extended set of experiments with heterogeneous traffic, variable packet sizes, and physical layer transmission rates we refer the reader to [12].

5.1. Experimental, simulation setups, and used performance metrics

Here we describe the common settings for all simulations (with the network simulator ns-2.27³) and the real-world experiments. In all setups we used TCP Newreno as the most popular variant of TCP. We do not use “window clumping” [15, 16], the mechanism that improves TCP performance in multihop wireless network. The idea behind CWND restriction is to not allow the particular TCP session to send more traffic than the network can handle. Apparently, this idea is embedded in our ingress throttling scheme. The source nodes in our scheme limit their transmission rate in order to not overload the bottleneck L-regions. Therefore additional rate limitation at the TCP layer is not necessary since the protocol will automatically adapt its CWND to the reduced “bandwidth.”

In the experiments presented in this article, we set the value of the maximum TCP data segment size (MSS) to 600 B. In all experiments except of the scenario with node mobility, the transmission rates at the physical layer is set to 2 Mb/s. In the latter case, the transmission rate between nodes equals 11 Mb/s. The 1 Mb/s transmission range is 250 m; the transmission range for 11 Mb/s is 30 m; the carrier sensing range (including the interference range) is 500 m.

We use FTP file transfers in both real-world experiments and simulations. 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 real-world experiments and simulations is 120 seconds.

In the experiments below we intentionally exclude traffic produced by an ad hoc routing protocol in order to focus on functional properties of our solution and not on an auto-configuration of the used parameters for our ingress throttling scheme. The evaluation of the impact of ad hoc routing on the improved TCP performance goes beyond the scope for this paper, we refer the reader to [17] for the detailed discussion on the topic.

In all experiments, we assume that the information about the bottleneck C-load shares, the physical layer transmission

³ Available online at <http://www.isi.edu/nsnam/ns/>.

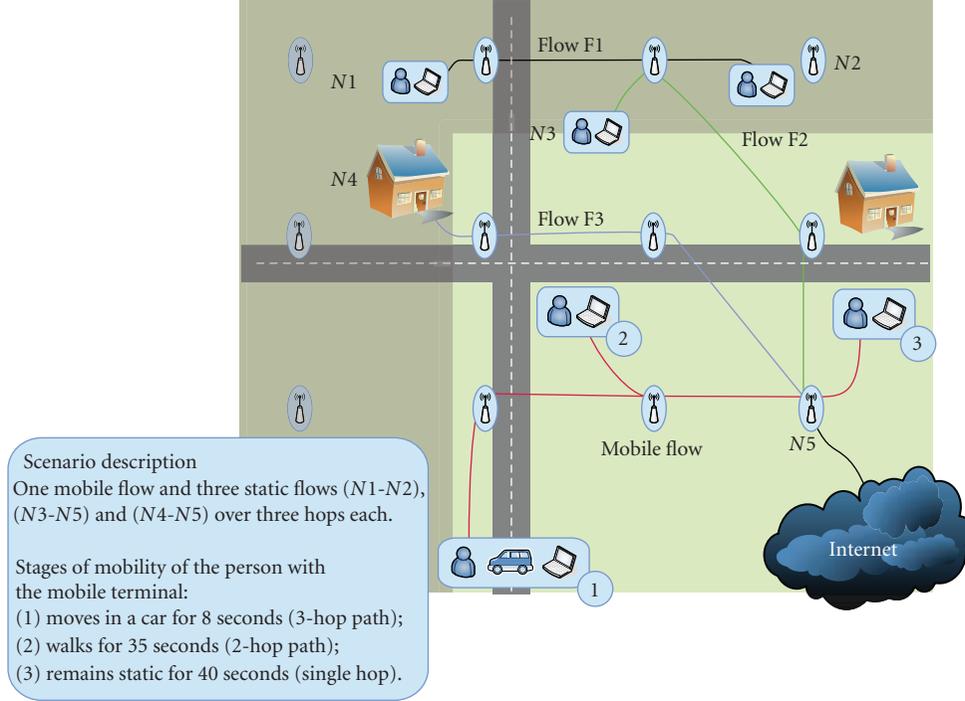


FIGURE 5: Network setup for the mobility experiment.

rates on the path, and the estimates of the ideal throughput for the flows with corresponding parameters is available at sources of TCP flows. The value $\text{Thr}_{\max}(h, \text{MSS}, \text{TX}_{802.11})$ needed to compute the delay parameter Δ (3) of the scheduler at the interface queue is obtained as is described in Section 4.3. In the experiments on static topologies we configure the delay parameter of the scheduler prior to the start of the experiments. As for the mobility experiment, the pre-orchestrated scenario described below allows us to deterministically decide on the handover times. During the simulation run at a time when the handover should occur we instruct the simulator to configure delay parameter of the scheduler according to the current network conditions.

5.1.1. Performance metrics

In the experiments we assess the network performance using the following set of metrics.

- (1) Individual (per-flow) TCP throughput. Denote this metric as Thr_i where i is the index of the particular TCP connection.
- (2) Combined (total) TCP throughput of all existing in the network TCP flows. Denote this metric as Thr_{tot} .
- (3) Unfairness index u : it is the normalized distance (4) of the actual throughput of each flow from the corresponding optimal value

$$u = \frac{\sqrt{\sum_{i=1, \dots, n} (\text{Thr}_{\text{opt}_i} - \text{Thr}_{\text{act}_i})^2}}{\sqrt{\sum \text{Thr}_{\text{opt}_i}}}. \quad (4)$$

In this formula $\text{Thr}_{\text{opt}_i}$ 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 computed for a particular scenario to the flow's throughput obtained when running alone in the network. $\text{Thr}_{\text{act}_i}$ 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.

5.2. Performance gains due to ingress throttling in the case of mobility

We begin the performance assessment of our ingress throttling first by considering a scenario with node mobility. Since evaluating the TCP protocol in sophisticated scenarios with complex mobility models would be too ambitious task for this paper we concentrate on a simple but nevertheless illustrative scenario. As a show case, we consider the network depicted in Figure 5. We have an ad hoc network where most of participants are relatively static. The scale of the topology is realistic and is chosen assuming 11 Mb/s transmission rate at the physical layer between the neighboring nodes. The inter-node distances between the communicating nodes is 30 m. In this topology most wireless routers are able to communicate with each other at least with the base IEEE 802.11b transmission rate 1 Mb/s. In this simple scenario one mobile

TABLE 1: The effect of rate throttling in the case of node mobility.

	Throughput at corresponding receivers, [kb/s]							
	Before first handover		After first handover		After second handover		Average throughput	
	Original perf.	Ingress throttling	Original perf.	Ingress throttling	Original perf.	Ingress throttling	Original perf.	Ingress throttling
Mobile flow	169.8	196.2	177.3	288.3	338.8	566.4	228.6	350.3
Flow F1	336.0	284.3	241.7	196.6	234.2	196.6	270.6	225.8
Flow F2	221.2	238.9	174.3	196.3	197.0	196.6	197.5	210.6
Flow F3	189.3	209.0	233.9	196.6	242.0	196.6	221.7	200.7

node maintains a data flow towards the Internet which initially (position 1 in Figure 5) runs over three wireless hops, then two and finally one hop at position 3. During the whole duration three other *static* flows (Flow F1 from $N1$ to $N2$, Flow F2 from $N3$ to $N5$, and Flow F3 from $N4$ to $N5$) compete with the mobile flow. All static flows follow a path of three hops. A summary of the scenario is given in the figure. The data flows are activated one after another with 2 seconds interval starting from flow F1, then flows F2 and F3 and finally the mobile flow.

The results of our mobility experiment are summarized in Table 1.⁴ One could intuitively expect that as soon as the mobile flow hands over to a shorter path, its throughput would increase correspondingly. However, this is not the case for the plain combination of TCP and IEEE 802.11. Apparently after the first handover, the mobile flow does not benefit from the two hops communications at all. This is because of the intensive interference coming from the static background flows.

When we enable our ingress throttling, the mobile flow has an opportunity to transmit faster when switching to shorter paths. At the same time the faster transmission of the mobile session does not harm the competing static flows. Their throughputs do not go below the fair share (196.6 kb/s). Moreover, we observe a 6% increase in the throughput for flow B in comparison to the case without throttling. This is because in our architecture the mobile flow—although it transmits with a higher rate—does not use more *air time* than the competitors. Overall, we observe nearly 8% increase in average total network throughput as well as perfect fairness (which is only partially visible in Table 1).

5.3. Scaling up network size and competition

This time we study the effect of the fairness framework and the ingress throttling scheme considering a set of experiments covering scenarios of increasing complexity. We scale up the network in two dimensions: the lengths of connections and the numbers of competing flows. The network set-

ting is shown in Figure 6(a). Since the unfairness metric is valid for evaluation of two or more flows we varied the number of competing TCP flows from 2 to 9. The route lengths for each flow is scaled from 1 to 9 hops. The results are summarized in Figures 6(c) and 6(d). The three-dimensional surfaces show the dynamics of unfairness index for a wide range of topologies. From Figure 6(c) it is visible that unfairness manifests itself even in simple scenarios: the unfairness in the case of six one-hop flows (i.e., 12 nodes network in general) is more than 10%. If we consider more complex formations, the situation becomes much worse. In the case of three hops networks (the part of the surface marked by a bold curve) the network behaves unstably and the unfairness peaks at 50%. From Figure 6(d) we observe that the unfairness virtually vanishes even in large networks with high number of competing connections when throttling according to our rate limit is implemented.

In this experiment, we can also demonstrate the validity of our hypothesis of taking the maximal throughput of a single TCP connection as a reference to the boundary C-load inside an L-region. Consider a subset of the topologies in Figure 6(a) where all competing flows follow a path of three hops (the dynamics of the unfairness index for these networks is marked by bold curves in Figures 6(c) and 6(d)). We can show that in this case the majority of network nodes are located inside a single common bottleneck L-region. For these networks we measure a combined TCP throughput achieved by all competing flows (Thr_{tot}). In addition to this we also measure a TCP throughput of a single TCP session when it does not compete for the radio medium with other flows. Indeed, if our hypothesis is wrong, then the total TCP throughput of all TCP connections running without shaping will be higher than this value.

As we observe from Figure 6(b), the total TCP throughput in the case where no shaping is done by the sources is always lower than the throughput of a single session (the straight line marked “Estimated” in the figure). By this we confirm our hypothesis from Section 4 to consider TCP throughput of a single TCP flow in isolation as a reference to the boundary C-load of the L-region.

From the figure, we observe that enabling the ingress throttling the resulting total throughput is equal to or larger than that in the case of plain combination of TCP and the IEEE 802.11 MAC. Moreover, the maximal deviation from the estimated value in the case where our scheme is enabled is only 3%, while in the case without throttling this value

⁴ Note that the fair throughput for the three hops flows in the case where all flows are active is 196.6 kb/s. The higher values for the throughputs of Flows F1, F2, and F3 in the column “Before first handover”—“Ingress throttling”—are due to different starting times for each flow.

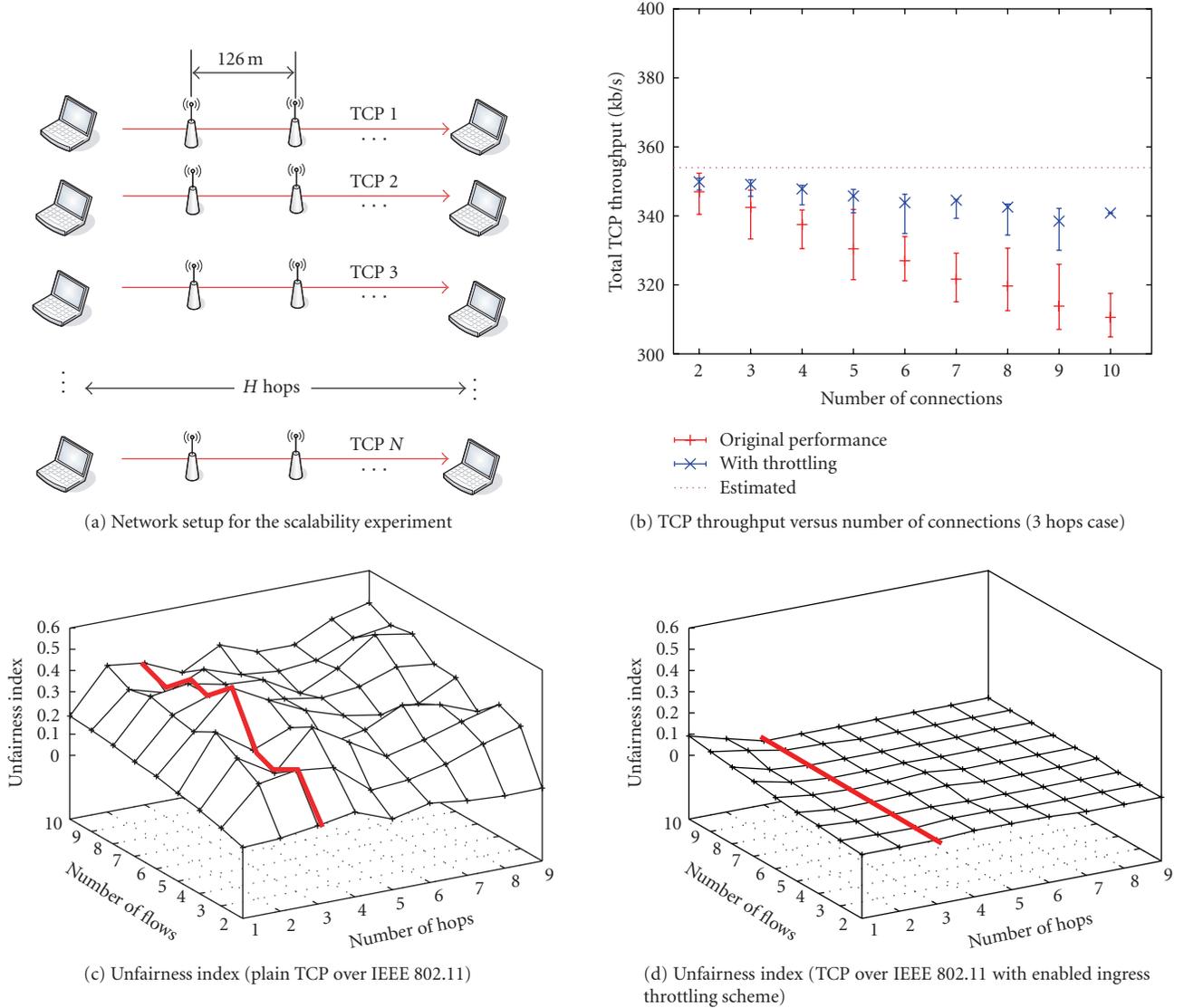


FIGURE 6: TCP unfairness index and TCP throughput (simulations).

is 12%. The reason for that we cannot achieve a perfect match of the total throughput to the estimated value is that controlling the load inside the L-region, we do not control the contention at the MAC layer. As a result, packets transmitted simultaneously from different stations collide during transmission. Therefore each individual and subsequently the total throughput in the network decrease.

5.4. Performance gains due to ingress throttling in a multiple bottlenecks network

In this experiment, we assess our fairness model by considering networks with multiple bottleneck L-regions. We use the topology depicted in Figure 7(a). In the network we have four TCP connections with different path lengths, the internode distance is 126 m. All flows use the same transmission rate at the physical layer -2 Mb/s and generate pack-

ets of equal size -600 B. In this network, we can identify three bottleneck L-regions with three competing connections (TCP1, TCP2, and TCP3) and four bottleneck L-regions with two competing connections (TCP1 and TCP4). For simplicity of the presentation, only one bottleneck of each kind is marked in the figure. Applying the algorithm of C-load shares distribution: $\phi_{TCP1} = 1/3$, $\phi_{TCP2} = 1/3$, $\phi_{TCP3} = 1/3$, and $\phi_{TCP4} = 2/3$. Thus L-region 1 is the bottleneck for flows TCP1, TCP2, and TCP3 and L-region 2 is the bottleneck for flow TCP4, therefore the allocation of C-load shares is max-min fair. We run two sets of simulations with enabled and disabled ingress throttling. Figure 7(b) shows the results from this experiment.

The first bars in each group show the estimated ideal throughput for each flow. We can immediately mark a severe TCP capture with respect to TCP1 in the case where all flows run over standard IEEE 802.11b network. Comparing the

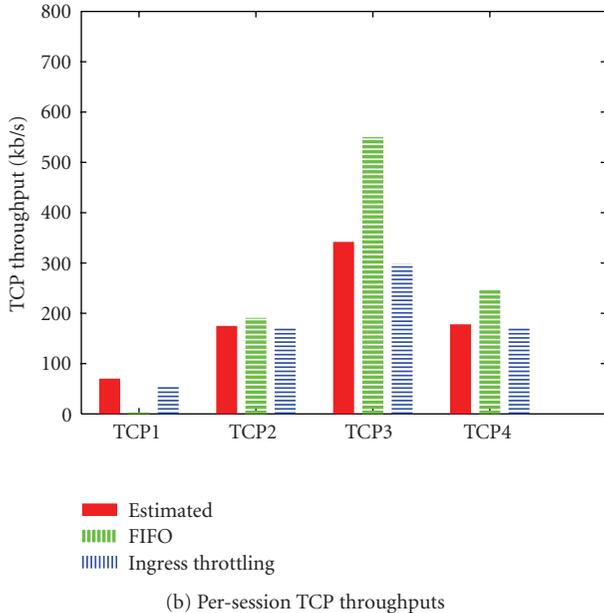
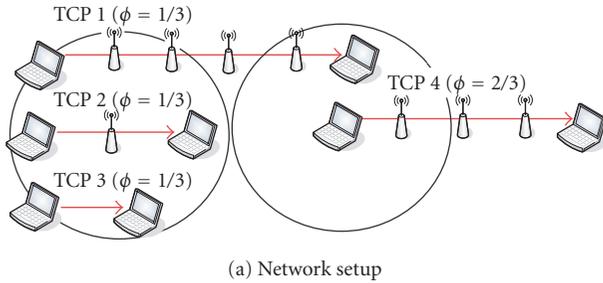


FIGURE 7: Network setting for the multiple bottleneck experiment and performance results.

resulting TCP throughputs in this case with the estimated ideal value, we see that the almost complete shut down of TCP1 is caused by the joint effect of transmissions from TCP3 and TCP4. Being the shortest of all flows, TCP3 transmits faster. Apparently the congestion control of TCP2 is able to capture its share of load in L-region 1. However TCP1 competes not only in L-region 1 but also in L-region 2. Inside L-region 2, TCP4 has a very favorable situation—it competes only with TCP1 which is already weakened by the competition with TCP2 and TCP3. In this situation, it is not a surprise that TCP4 makes the situation for TCP1 even worse.

Now, let us have a look at the bars corresponding to the case where all flows are aware about the competitors on the path and throttle the output rate of TCP segments according to the assigned shares (see right bars in each group). We observe that in this case the communications are fair with respect to each flow. In this case each source generates as much load as the corresponding bottleneck L-region can handle. By doing this no one flow is able to capture the capacity.

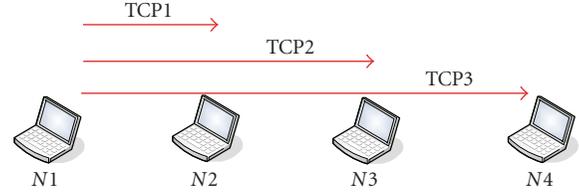


FIGURE 8: Network setup for the real-world experiment.

TABLE 2: The effect of rate throttling in the case of flows with variable lengths.

	Simulations		Real-world test	
	Original perf.	Rate throttling	Original perf.	Rate throttling
Thr _{tot} (kb/s)	554	605	659	667
Unfairness	0.31	0.09	0.27	0.04

5.5. Comparative assessment of the fairness framework in real-world testbed and simulations

The purpose of this experiment is to assess our findings not only in simulations but also in a real-world testbed. As a showcase we choose the topology depicted in Figure 8. The three competing flows follow the paths of one, two, and three hops, respectively. The physical testbed consists of four DELL Latitude laptops with ZyXEL ZyAIR B-100 wireless interfaces. We use the Linux operating system with 2.6 kernel and Lunar⁵ ad hoc routing to set up multihop paths. Since physically the laptops were located in the same room, the multihop connections were achieved with the help of the *mackill* utility of Lunar, which forbids not topologically neighboring nodes to answer the route request messages. We used the default transmission power of B-100 cards and configure the transmission rate at 2 Mb/s. At the source node $N1$, prior to the start of the experiment we configured traffic controller *tc* supplied with the Linux distribution to sort packets belonging to different flows into separate logical queues. We also measured the values of maximum throughput for each competing flow in isolation and precomputed the delay parameter for each logical queue. As soon as the path for each connection is established, we commanded the *tc* utility to delay packet transmission in each queue with the corresponding delay value. The data traffic was generated by three FTP transfers of a large file from node $N1$ requested by nodes $N2$, $N3$, and $N4$, respectively.

We perform this experiment both in the simulator and in a real-world testbed. We measure the combined TCP throughput (Thr_{tot}) and the unfairness index u . Table 2 shows the resulting performance in this experiment. We observe that the total TCP throughput is increased on 50 kb/s

⁵ Uppsala University ad hoc implementation portal. Available online at http://core.it.uu.se/core/index.php/Main_Page.

in simulations and 10 kb/s in real-world experiment. The unfairness reduced drastically more than 3 times in simulations and almost 7 times in the testbed. Although the exact values of the used metrics are different in simulations and the real-world experiments, in this experiment we show a similar performance improvements dynamics in the two used environments.

6. RELATED WORK AND OPEN RESEARCH ISSUES

Adaptation of the transmission rate for improving the fairness in MANETs was suggested in [5, 6]. In the first paper the max-min fairness framework is formulated for the case of MANETs. The second paper extends the model to address *proportional* fairness. In both papers, the capacity in MANETs is considered as a function of space, which is a common part with our max-min fairness framework.

The major difference between these two fairness models and our work is the targeted implementation place in the TCP/IP protocol stack. The framework in [5, 6] is intended for implementation at the MAC layer. The authors view a single multihop session as a set of independent one hop connections and distinguish between them by overhearing the transmissions of MAC layer control messages and data packets. The fair distribution of the capacity between the competing one hop connections is done by a special message exchange scheme. The transmission rate of each one hop flow is conformed to the assigned capacity share by adjusting the contention window at the MAC layer depending on the measured offered load. As for our *ingress throttling* scheme, we designed it relying on session's parameters available only before the MAC layer. We compute the capacity shares in bottleneck regions on the *per-IP-session* basis and propagate this information to other end-to-end connections ongoing beyond the reachability of the MAC protocol for the particular node. The rate limitation is performed at ingress nodes only, the intermediate nodes only execute the *share assignment* algorithm.

A cross-layer approach based on the network layer feedback called ATCP is proposed in [18]. The information that the network feedback mechanism reports to the sources of TCP connections is (1) failure of the route; (2) the event of a packet loss during the transmission on MAC layer; and (3) the true network congestion. In all these cases TCP is instructed to adapt its behavior as following. On the loss of the connectivity the TCP sender goes into a persistent state, so that it does not unnecessarily (re-)transmit packets. As a reaction to bit error related packet loss, the ATCP layer retransmits the lost packet before invocation of the congestion control mechanism by TCP. Finally, the congestion control performs the traditional actions in the case of a network congestion.

In [7] the authors introduce a new congestion control algorithm for TCP over multihop IEEE 802.11-based networks. The core of the suggested algorithm is an adaptation of the transmission rate of TCP at the sending node using

current estimation of the end-to-end delay and the coefficient of variation of recently measured round trip times. In the evaluation of the proposed scheme, the authors account both for throughput and fairness characteristics of TCP connections and show significant improvements of these characteristics in comparison to the related approaches.

Yang et al. [19] propose to limit the arrival rate from the interface queue to the transmission buffer on MAC layer depending on the observations of the departure rate in the past. Their proposal assumes that the rate limitation at a transmitting node is implemented at the link layer, thus leaving the functionality of the MAC protocol unchanged. The authors compute the transmission rate limits using past observations of the packet emission rate from the node and apply heuristics for the rate control. The control mechanism is implemented in forwarding nodes as well as in sources of communications.

6.1. Open research issues

Having outlined the major concepts of MANETs fairness framework and mechanisms of the ingress rate control, we highlight in this section some open questions and problems for further investigation.

6.1.1. Formal estimation of the maximally achievable TCP throughput in MANETs

In the first place we suggest that further developments of the proposed architecture should be directed on formal analysis of the maximally achievable throughput of single TCP session. A few papers attempt to formally model a single TCP flow over multihop wireless network. In [20] the authors formally characterize the throughput of a single multihop TCP connection in wireless network. The work describes an approach to model TCP transmissions over IEEE 802.11 network with enabled RTS/CTS exchange as an embedded Markov chain.

6.1.2. Excess capacity and utilization

In our scheme, the particular end-to-end data session will be guaranteed a share of the *available* network capacity for its entire duration considering its worst case communication, namely when it always has data to transmit, for example, long file transfer. If this is the case, our scheme provides both perfect fairness and good network utilization. However, in reality we will also have a number of short-living communications and flows with a number of short transmission bursts, for example, web browsing. In this case, the share of capacity will not be fully utilized. When several connections originate from the same source, the leftover capacity can be reused by the active flows, otherwise the network will be underutilized until the route for the inactive connection will be removed. Therefore it is essential to have an effective dynamic mechanism to update the ingress nodes with information of the C-load usage.

6.1.3. Wireless stub and transit networks

So far we considered a pure MANET scenario where we looked at TCP flows starting and terminating inside the wireless network. We need to extend our study to mesh networks where TCP flows (i) start in MANET and end in fixed network, (ii) start in fixed network and end in MANET or (iii) use MANET clouds as transit networks.

The first case corresponds to the normal functionality of the described solution. In the last two cases, our functionality will naturally be added to the ingress gateways of MANET. While in case (ii) the ingress node should place packets of distinct connections in a separate queue, in case (iii) queuing should be done on an ingress-egress aggregate basis.

6.1.4. Towards autonomic ad hoc network architectures

Our resource protection layer provides an adaptive and stable but conservative operation point. As the issues raised above show, it can be optimized depending on different circumstances. We believe that determining a “universal” resource management protocol for all possible cases will not be possible. Instead, we favor an “autonomic networking” approach where nodes actively try out relaxations of our scheme at run time to permit better resource utilization. If problems arise, the system can revert back to the stable operation point. Only with such self-monitoring and self-optimizing properties can a wireless network become adaptive in a broad sense and achieve the dream of ubiquitous ad hoc networking.

7. CONCLUSION

In this article, we considered the problem of the severe unfairness between multiple multihop TCP flows in a wireless ad hoc network. We adapted the max-min fairness framework to the specifics of the wireless environment. We described an ingress throttling scheme which enforces the fairness model. Our solution is an adaptive distributed capacity allocation scheme for multihop wireless networks. The capacity is allocated on a per-session basis at a specific point in time during the route establishment.

We showed with simulations as well as with real-world experiments full compliance of the *ingress throttling* scheme to our *space-load* fairness model. The major result of this article is that by limiting the transmission rate at sources of TCP flows according to the derived rate limiting formula, TCP capture does not occur, the total throughput is increased, and almost perfect fairness is achieved for all involved flows.

REFERENCES

- [1] K. Xu, S. Bae, S. Lee, and M. Gerla, “TCP behavior across multi-hop wireless networks and the wired Internet,” in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM '02)*, pp. 41–48, Atlanta, Ga, USA, September 2002.
- [2] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, pp. 219–230, Seattle, Wash, USA, August 1999.
- [3] Y. Tian, K. Xu, and N. Ansari, “TCP in wireless environments: problems and solutions,” *IEEE Communications Magazine*, vol. 43, no. 3, pp. S27–S32, 2005.
- [4] Ch. Tschudin and E. Osipov, “Estimating the ad hoc horizon for TCP over IEEE 802.11 networks,” in *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net '04)*, Bodrum, Turkey, June 2004.
- [5] X. L. Huang and B. Bensaou, “On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 221–231, Long Beach, Calif, USA, October 2001.
- [6] L. B. Jiang and S. C. Liew, “Proportional fairness in wireless LANs and ad hoc networks,” in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 3, pp. 1551–1556, New Orleans, La, USA, March 2005.
- [7] S. M. ElRakabawy, A. Klemm, and C. Lindemann, “TCP with adaptive pacing for multihop wireless networks,” in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 288–299, Urbana-Champaign, Ill, USA, May 2005.
- [8] P. Bahl, R. Chandra, and J. Dunagan, “SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 216–230, Philadelphia, Pa, USA, September-October 2004.
- [9] E. Osipov and Ch. Tschudin, “A path density protocol for MANETs,” *Ad Hoc and Sensor Wireless Networks (Old City Publishing)*, 2007.
- [10] G. Anastasi, M. Conti, and E. Gregory, *IEEE 802.11 Ad Hoc Networks: Protocols, Performance and Open Issues*, John Wiley & Sons, New York, NY, USA, 2004.
- [11] J.-Y. L. Boudec, “Rate adaptation, congestion control and fairness: a tutorial,” EPFL, February 2005. http://ica1www.epfl.ch/PS_files/LEB3132.pdf.
- [12] E. Osipov, “On the practical feasibility of fair TCP communications in IEEE 802.11 based multihop ad hoc wireless networks,” Ph.D. dissertation, University of Basel, Basel, Switzerland, 2005.
- [13] B. Awerbuch, D. Holmer, and H. Rubens, “High throughput route selection in multi-rate ad hoc wireless networks,” in *Proceedings of the 1st IFIP TC6 Working Conference Wireless On-Demand Network Systems (WONS '04)*, pp. 253–270, Madonna di Campiglio, Italy, January 2004.
- [14] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM '03)*, pp. 134–146, San Diego, Calif, USA, September 2003.
- [15] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, “The impact of multihop wireless channel on TCP throughput and loss,” in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1744–1753, San Francisco, Calif, USA, March-April 2003.
- [16] M. Gerla, K. Tang, and R. Bagrodia, “TCP performance in wireless multi-hop networks,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications (WMCSA '99)*, pp. 41–50, New Orleans, La, USA, February 1999.

- [17] E. Osipov and Ch. Tschudin, "Evaluating the effect of ad hoc routing on TCP performance in IEEE 802.11 based MANETs," in *Proceedings of the 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN '06)*, pp. 298–312, St. Petersburg, Russia, May-June 2006.
- [18] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, 2001.
- [19] L. Yang, W. K. G. Seah, and Q. Yin, "Improving fairness among TCP flows crossing wireless ad hoc and wired networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 57–63, Annapolis, Md, USA, June 2003.
- [20] A. A. Kherani and R. Shorey, "Throughput analysis of TCP in multi-hop wireless networks with IEEE 802.11 MAC," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 1, pp. 237–242, Atlanta, Ga, USA, March 2004.