A Simulation Study of XCP-b Performance in Wireless Multi-hop Networks

Filipe Abrantes
INESC Porto, Faculdade de Engenharia,
Universidade do Porto
Rua Dr. Roberto Frias, 378, Porto, Portugal
fla@inescporto.pt

Manuel Ricardo
INESC Porto, Faculdade de Engenharia,
Universidade do Porto
Rua Dr. Roberto Frias, 378, Porto, Portugal
mricardo@inescporto.pt

ABSTRACT

XCP-b proposes a modification to the XCP router algorithm that computes the spare bandwidth. The modification removes the need for an XCP router to know the exact capacity of the channel, making it possible to use the XCP-b variant in transmission media where the capacity is hard to measure. An example of this kind of medium is the IEEE 802.11. Previous work shows that XCP-b behaves well in single-hop wireless networks and that it outperforms TCP in terms of fairness, queuing delay, stability and efficiency when the bandwidth delay product of the network grows. In this paper we extend the validation and evaluation of XCP-b to the case of multi-hop wireless networks, both stand-alone and as access networks to other wired networks.

The results show that XCP-b maintains its fundamental characteristics in wireless multi-hop scenarios, such as stable throughput and low standing queues, while distributing the bandwidth fairly and using the available capacity efficiently. The simulations also show that XCP-b produces congestion window values that are closer than TCP to the theoretical upper-bound which maximizes spatial reuse.

Categories and Subject Descriptors

C.2.2 [**Computer Communication Networks**]: Network Protocols

General Terms

Performance

Keywords

congestion control, dynamic bandwidth, XCP, wireless multi-hop, mesh.

1. INTRODUCTION

Van Jacobson's algorithm [10] used by TCP [16] has been the main responsible for congestion control in the Internet since the early 90's. Although it has performed its task remarkably well, it has some characteristics that degrade the QoS of the network. Such characteristics include increased queuing delay, unstable throughput, limited fairness and slow and inefficient behaviour in high bandwidth delay product networks. The root of the problem is the fact that TCP's algorithm relies on rare events which carry low resolution information. This leaves too much guessing to be done by TCP, resulting in TCP making inaccurate decisions.

XCP [12] proposes a new approach, where routers take an active role in informing sources about the state of the network, and how to adjust the sending rate, giving precise feedback. XCP flows have a stable throughput and achieve fair bandwidth allocation, the bottleneck capacity is fully utilized for any bandwidth delay product network, and the bottleneck queue stabilizes at zero length. In order to calculate the precise feedback for each source, the XCP controller needs to compute the current spare bandwidth of the link, and for that the XCP controller needs to be configured with the capacity of the outgoing link. In shared-access multi-rate media, knowing the exact capacity of the link and the fair share of each station is difficult, thus XCP struggles to work in such media.

XCP-b is an alternative to XCP, which is better suited for these media where the capacity is not easy to measure. It proposes that the spare capacity of the link is computed by measuring variations of the persistent queue of the outgoing link. The XCP-b algorithm has been validated in single-hop wireless scenarios and showed that XCP properties are maintained: stable throughput, low queuing, accurate fairness, and high efficiency as the bandwidth delay product scales.

In this paper we extend the evaluation of XCP-b to scenarios including wireless multi-hop scenarios both functioning as access networks or stand-alone networks. The paper is organized as follows. § 3 describes briefly the operation of XCP, the effect of capacity estimation errors, and the modifications proposed by XCP-b. § 4 elaborates on the problems of typical congestion control in multi-hop wireless networks. § 5 describes the simulation scenarios and presents the performance results. In § 6 we present the conclusions of the paper and discuss future work.

2. RELATED WORK

In recent years, congestion control in wireless multi-hop networks has been the topic of several studies, including [9], [8], [5], [13], [6], [14]. These studies focus on the interaction between TCP and the 802.11 MAC layer: the prob-

Q2SWinet'07, October 22, 2007, Chania, Crete Island, Greece.

lems of self-interference, capture, high latency, unfairness, and capacity limits. In [8], it is noted that exists a value of the congestion window that maximizes throughput, by maximizing spatial reuse. Beyond this value, collisions and interference start to reduce the capacity. This value of the congestion window is generally much lower than the one at which TCP Newreno operates. [5] supports these results by showing that exists an upper-bound to the bandwidth delay product of a multi-hop wireless network. These studies provide a good source for comparison of the congestion window obtained by the congestion control algorithms, however, they only apply to the case of stand-alone multi-hop wireless networks.

In [17] the authors present an alternative XCP router feedback algorithm which can also be used in 802.11 networks. This algorithm differs from ours in the sense that it uses MAC level information, while we only rely on the queue dynamics.

In this paper we extend the validation and performance study of XCP-b to multi-hop wireless networks. We only give a brief explanation of the algorithm in this paper, for a more detailed description the reader should refer to [3], [12], and [7].

3. XCP AND XCP-B BACKGROUND

XCP is a protocol that enables queue controllers in a path to inform sources how to adjust their sending rates. This communication between sources and the network is enabled by using an header between the network and transport headers. The XCP header, or congestion header, carries the flow throughput and RTT values, and a field where queue controllers fill the adjustment required to the flow's throughput in the next interval. The adjustment is calculated on a per-packet basis.

3.1 XCP algorithm

The calculation of the adjustment required to a certain flow is split in two algorithms: the efficiency algorithm, and the bandwidth allocation algorithm. The efficiency algorithm periodically (every d seconds) calculates the amount of bandwidth F that will be distributed among all flows during the next d seconds:

$$F = \alpha \cdot (C - input\_bw) - \beta \cdot \frac{q}{d} \tag{1}$$

where C is the capacity of the link, $input\_bw$ is the bandwidth actually used during the last period d, and q is the persistent queue or, in other words, the minimum queue length observed during the last d seconds. d is usually set to be the average RTT of the flows traversing this queue. α and β are constants.

The bandwidth allocation algorithm will then distribute F among the currently active flows following an AIMD rule. If F is negative, each flow is decremented proportionally to its rate, otherwise F is distributed equally among all flows. The detailed algorithm is described in [7].

3.2 Capacity estimation errors

Choosing a capacity value C for the computation of F (Eq. 1) is difficult when the medium used is 802.11. C depends on the data rates used by each station, the number of active stations, the number of collisions, and the handshake mechanisms and thresholds. Previous studies [18], [3] showed that XCP is able to compensate errors in the estimation of the capacity by building up the queue. The queue length required to compensate the error is proportional to the error ϵ itself and to the average RTT of the flows d. For low estimation errors ($\epsilon < 0.1 \cdot C$) it can be approximated by [3]:

$$q \simeq \frac{\alpha}{\beta} \cdot \epsilon \cdot d \tag{2}$$

For higher estimation errors ($\epsilon \geq 0.1 \cdot C$) we need to introduce the effect that the increase of the queuing delay has on d. In this case, the queue length required to compensate the error becomes:

$$q = \frac{\alpha}{\beta} \cdot \epsilon \cdot \sum_{i=0}^{i=\infty} \left(\frac{\alpha}{\beta} \cdot \frac{\epsilon}{C} \right)^i \cdot d_{base} \tag{3}$$

which can be simplified into:

$$q = \frac{\alpha}{\beta} \cdot \epsilon \cdot \frac{1}{1 - \frac{\alpha}{\beta} \cdot \frac{\epsilon}{C}} \cdot d_{base}, \quad \epsilon < \frac{\beta}{\alpha} \cdot C \tag{4}$$

where d_{base} represents the average RTT of the flows with no queuing delay.

In 802.11 networks the error ϵ may range from 0 to the maximum capacity of the channel. The queue length required to compensate the estimation error may be too large, or it may not stabilize at all if $\epsilon > \frac{\beta}{\alpha} \cdot C$.

3.3 XCP-b variant

XCP-b(lind) proposes modifications to the calculation of F in order to avoid large queues due to capacity errors in shared-access media such as 802.11. Instead of calculating the spare bandwidth as the difference between the total capacity and the used capacity, XCP-b measures the spare bandwidth from variations of the persistent queue. The biggest drawback of this approach is that a queue controller can only measure queue variations when the medium is being fully utilized, so the system must be taken somehow to full utilization in order to work. When the queue cannot be measured, because the medium is under-utilized, XCP-b uses a fixed value for F. This fixed value is such that the queue length in the next control intervals never exceeds its limit Q_{max}. In order to minimize the oscillations between the two types of feedback – feedback based on queue variations, or fixed feedback – XCP-b maintains an exponential average of the persistent queue length, and it only switches between the two strategies when the exponential average crosses a certain threshold. Using a threshold in the exponential average of the queue allows XCP-b to ignore periods of empty queue due to the medium access randomness or during transitory periods of the queue response. The calculation of F proposed by XCP-b comes as:

$$F = \begin{cases} \chi \cdot \frac{Q_{max}}{d} & \text{if } \lambda < \tau \cdot \kappa, \\ -\alpha \cdot \frac{\Delta q}{d} - \beta \cdot \frac{q - \kappa}{d} & \text{if } \lambda \geq \tau \cdot \kappa. \end{cases} \tag{5}$$

where χ is a constant that controls the amount of the fixed value of F in under-utilization periods, κ is the value at which XCP-b tries to stabilize the queue length and τ is a constant that controls the lowest value of the exponential average of the persistent queue that considers the medium fully-utilized. λ represents the exponential average of the

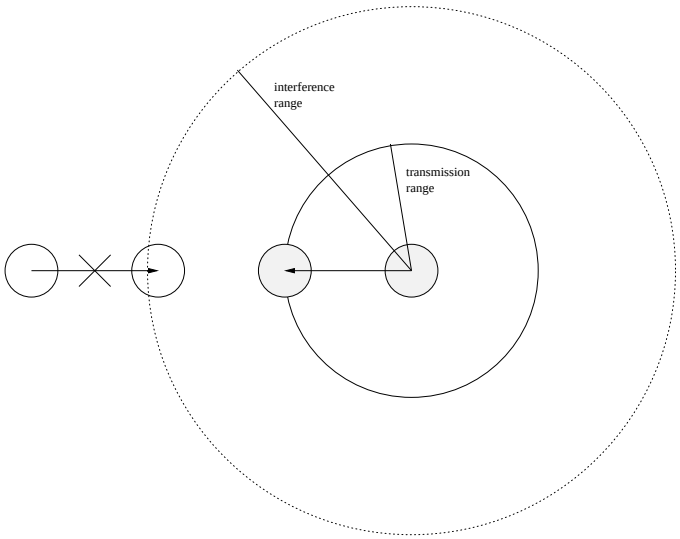


Figure 1: The hidden node and the interference problem.

length of the persistent queue. For a detailed explanation of the choice of the parameter values please refer to [3].

4. CONGESTION CONTROL ISSUES IN MULTI-HOP WIRELESS NETWORKS

Multi-hop 802.11 wireless networks have special characteristics that increase the network latency and unfairness, and decrease throughput. One of these characteristics is the problem of the hidden node [4]; a second characteristic is that the interference range is more than twice the transmission range. This may forbid nodes to transmit for long periods of time, if there is an on-going communication at interference range. Also, route breakages may occur, either due to mobility or loss of routing messages, which typically are broadcasted. Route breakages may cause consecutive losses, which induce longer retransmission timeouts (RTO), extending the time that a flow takes to resume communication. In this paper, we focus in the latency and unfairness, and consider static mesh networks.

Multi-hop wireless networks have high latency when the network is saturated. In this case the nodes compete simultaneously for the medium, the MAC contention increases, leading to queue build-up at the nodes and latencies which may be in the order of seconds. This increased latency not only affects interactive applications, but it also affects the dynamics of congestion control algorithms, whose pace of adaptation is tightly related to the RTT of the flow. The higher the latency, the slower the adaptation of the congestion control is. TCP and XCP(-b) are examples of algorithms whose dynamics are coupled with the RTT of the flows.

Another characteristic of multi-hop wireless networks is the increased packet loss probability. Even though the 802.11 MAC layer has reliable transmissions, which, in case of failure, can retransmit a packet up to 7 times, packet drops can still occur. This happens particularly because of the hidden node problem, and because the interference range is higher than the transmission range of the 802.11 technology. If communication is taking place in interference range, and it is occupying the channel for most of the time, several stations may have communication inhibited for long periods causing packet loss (Fig. 1). The losses, more than limiting the throughput of loss-based congestion control algorithms, interferes with the computation of the RTO, as consecutive losses may set an high RTO, causing the connection to be idle during this time. This results that nodes in "weaker" positions may starve. Even one single flow crossing more than 3 hops can experience losses caused by self-interference, as packets from the same flow will be traversing the same path and competing for the medium at the different nodes. The hidden node problem can be somewhat alleviated by the use of handshake mechanisms (RTS/CTS), however this usually comes at the cost network throughput.

Summing all up, bandwidth allocation to flows in multi-hop wireless networks may be extremely unfair, even causing starvation of some flows. Also, very large average delays and large delay variance may be experienced in saturated networks, which reduce the accuracy and pace of adaption of the congestion control algorithms. XCP-b does not address the 802.11 issues specifically, but by using a 16-bit value as primary feedback instead of losses in the network, XCP-b is able to control the sources accurately, producing low standing queues and precise fairness between flows. This allows XCP-b to minimize the effects that 802.11 characteristics have on the congestion control algorithm.

5. SIMULATION EXPERIMENTS

The purpose of the simulation experiments presented in this section is threefold. The first is to show that XCP-b maintains the properties presented in [3] in multi-hop networks. The second purpose is to measure the gains of XCP-b when compared to loss-based (Newreno) congestion control. The third is to understand the performance sensitivity of XCP-b to parameter tuning.

5.1 Simulations Setup

We use the ns-2 simulator [1], and the code used to run the simulations is available at [2]. We simulate the algorithms in stand-alone multi-hop chain topologies as well as in multi-hop wireless networks used as access networks to other wired networks. The scenarios we simulate do not include mobility. However, route changes may occur due to the loss of routing messages, or re-ordering of these messages in the network. We set the queue size of the wireless nodes to 25 *packet* and the 802.11 MAC is used. The XCP-b parameters are set as proposed in [3], except $Q_{max} = 16$ *kbyte* to match the average queue length in this setup (assuming the maximum queue length of 25 *packet*, and an average packet length of 650 *byte* - data packets are 1300 *byte* long and transport acknowledgements have 40 *byte*). The maximum RTO of TCP is set to 2 seconds to avoid large periods of connection idleness, as consecutive losses may be frequent. The data and basic rates of the stations are set to 11 and 1 Mbit/s respectively, unless stated otherwise. Also, no handshaking (RTS/CTS) or fragmentation mechanisms are used; some simulations were also done with RTS enabled, but they always resulted in significant throughput decrease due to the great overhead it represents. The results with RTS enabled are not shown here for the sake of space and visibility. The routing algorithm used is DSDV [15]. Sources are greedy and the packet size is set to 1300 *byte*, as stated before. The

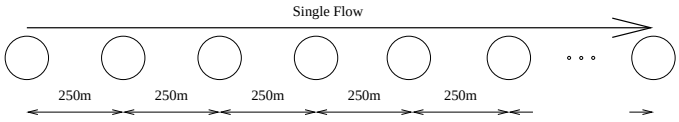


Figure 2: Chain Topology

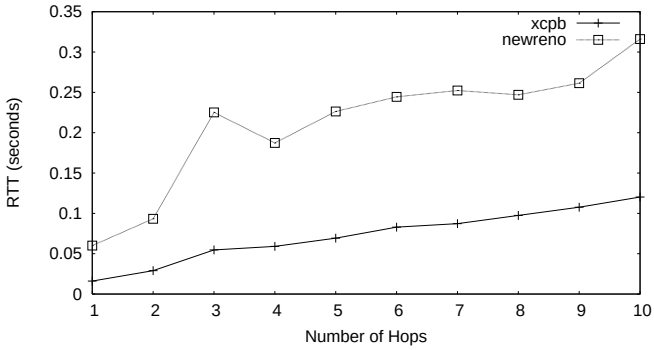


Figure 3: The average RTT of a single flow in a chain topology

TCP version used is NewReno without delayed ACKs, and ECN is not used. In these simulations we make two assumptions which might have some impact on the results: losses are caused only by medium collisions and not by transmission errors; sources have always packets to send, that is they are greedy.

5.2 Chain Topology

In this section we analyze the results obtained from experiments of single flows over simple chain topologies similar to the one represented in Fig. 2. We compare the congestion window, average RTT, and average throughput obtained by XCP-b and TCP Newreno. A previous study [5] shows that exists an upper-bound to the value of the congestion window, beyond which the increase of the congestion window will only increase the number of collisions, reducing efficiency. The upper-bound is $\approx \frac{1}{5}$ of the round trip hop count (RTHC). Comparing the results obtained by XCP-b and TCP Newreno, we observe that the average congestion window obtained by XCP-b is closer to the limit set in [5]. Still, it does set the congestion window above the limit, which happens because XCP-b nodes need to build-up some queue in order to calculate the spare bandwidth. Even though the stabilizing queue is small, it still adds some delay to the path, which causes the congestion window to grow beyond the limit. This drives the channel to the saturation point, thus the number of collisions is not optimal, resulting in no gain in terms of flow throughput relatively to TCP Newreno (Fig. 4). The queue build-up required by XCP-b does add some delay to the path, however, it still is much lower than the delay introduced by TCP as shown in Fig. 3.

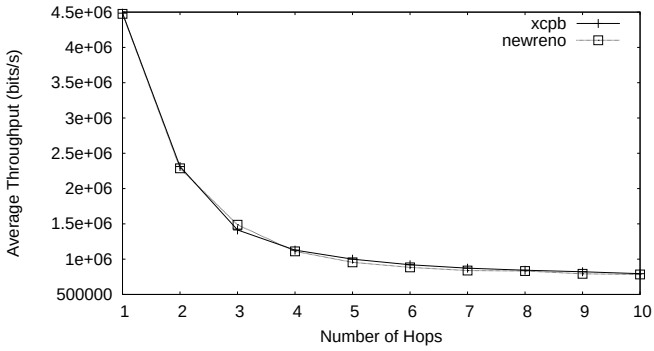


Figure 4: The average throughput of a single flow in a chain topology

Table 1: Average congestion window in a chain topology

	XCP-b	Newreno	Limit [5]
1-hop	6.39	24.24	2
2-hop	6.57	20.78	2
3-hop	7.30	33.45	1
4-hop	6.57	19.85	1
5-hop	6.81	21.14	2
6-hop	7.56	20.48	2
7-hop	7.58	20.34	3
8-hop	8.16	20.71	3
9-hop	8.71	20.02	3
10-hop	9.37	22.79	3

5.3 Mesh Access Networks

In this section we evaluate the performance of XCP-b in networks with a wireless mesh access network based on 802.11. The simulations are based on the scenario presented in Fig. 5, where flows start from the nodes $N(n)$ to each of the stations in the groups T1, T2, T3, B1, B2, B3, R1, R2, and R3. Each of these groups has 3 stations. Flows of stations from groups T1, B1, and R1 start at $t = 0$, from groups T2, B2, and R2 start at $t = 20$, while flows from stations of groups T3, B3, and R3 start at $t = 40$. The MAC used by the wireless nodes is 802.11 in DCF mode. We increase the capacity of the network by using a data rate of 22 Mbit/s, and a basic rate of 2 Mbit/s. This allows us to run the simulation with an higher number of flows, giving more confidence to the results. With lower capacity (data rate of 11 Mbit/s and basic rate of 1 Mbit/s) and a high number of flows, the congestion windows of all the flows have the size of 1 *packet*, making it difficult to analyze the congestion control mechanism performance. Because of the increased capacity we now double the queue maximum length to 50 *packet* and also double the value of Q_{max} in XCP-b to 32 *kbyte*.

The wired links between the $N(n)$ nodes and R have a capacity of 100 Mbit/s and a delay of 50 ms. The wired link between R and the base station BS has a capacity of 1 Gbit/s and 1 ms delay. All flows between the wired and the wireless nodes have to go through the base station BS. We run the simulations using XCP-b and TCP Newreno. To observe the impact of losses on the XCP-b algorithm we run the simulations with 2 different versions of the XCP-b algorithm:

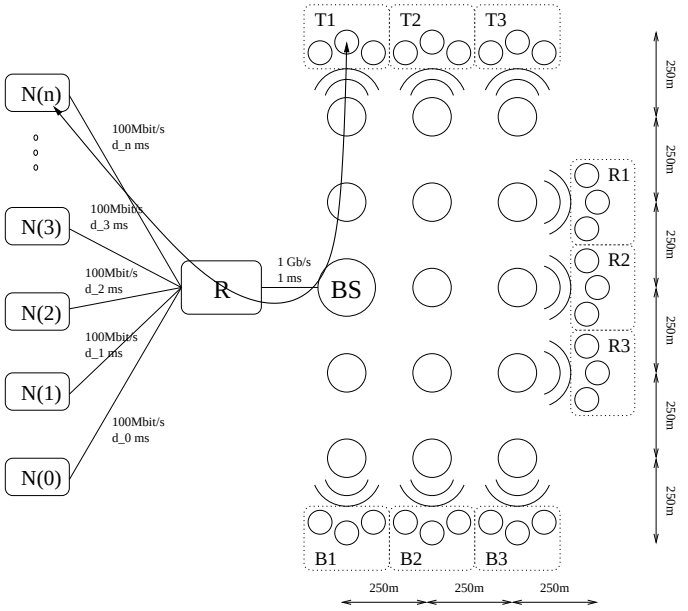


Figure 5: The scenario of a mesh-access network

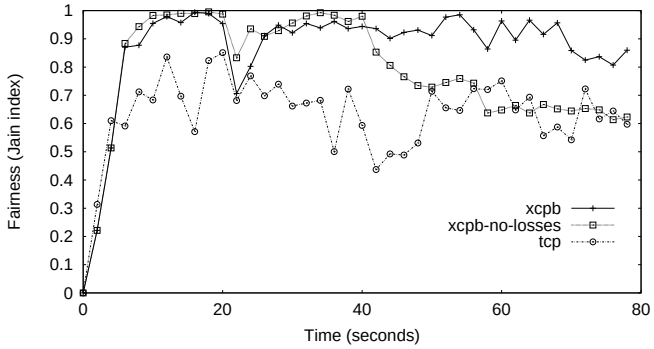


Figure 6: The Jain's fairness index of XCP-b, XCP-b with no reaction to losses, and TCP Newreno

one that does not react to losses, and other that maintains the behaviour of TCP to losses, that is, halving the size of the congestion window upon receiving three DUP-ACKs, and resetting the congestion window when the transmission of a packet times out. We analyze the performance of XCP-b and compare it to TCP Newreno in terms of fairness (Fig. 6), queuing delay (Fig. 7), throughput stability and total throughput (Table 2).

In terms of fairness, we measure the performance using the Jain's fairness index [11]:

$$J = \frac{\left(\sum_{i=1}^{n} \bar{x_i}\right)^2}{n \cdot \sum_{i=1}^{n} \bar{x_i}^2} \quad (6)$$

where $\bar{x_i}$ is the average throughput of source i, and n is the number of active sources during the interval considered to calculate the index value. In our experiment we consider periods of 1 second to calculate the index, and we only consider flows to be active if they are active during the full interval. For the sake of visibility the points shown in Fig. 6 are spaced by 2 seconds. The results show that XCP-b is

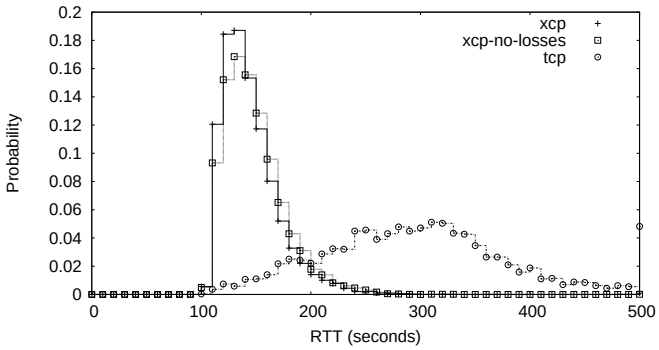
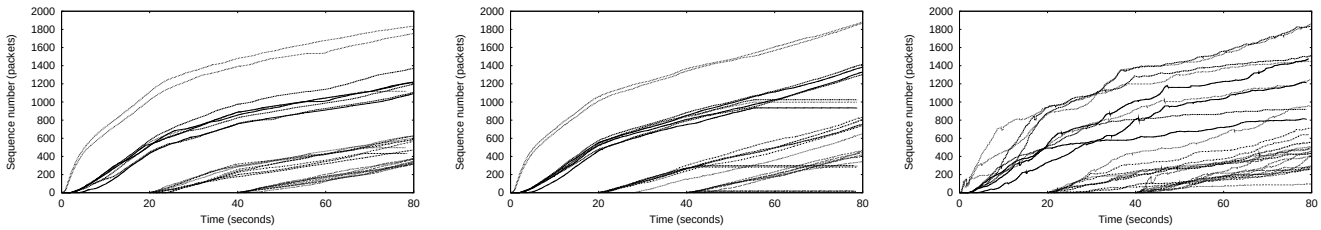


Figure 7: The probability density function of the distribution of the RTTs measured by the senders.

able to maintain near maximum ($J = 1$) fairness throughout most of the simulation time, while Newreno oscillates in the region of $J \in [0.4; 0.8]$. Notice the instants $t = 20$, and $t = 40$, in which new flows enter the network, which cause a disturbance in the fairness index, until the rates converge to fairness. One interesting fact, and maybe counter intuitive, is that XCP-b with no reaction to losses (XCP-b-nl) may actually perform worse than XCP-b which halves or resets the congestion window when losses are detected. This happens because XCP-b-nl maintains the network always near the saturation point of capacity; when flows starve, they find it difficult to regain bandwidth in the medium because it is always occupied, either by 1-hop neighbours or by interference from 2-hop neighbours. This is observable in Fig. 8(b) where several flows simply stall, as the competition for the medium increases. From this point of view, XCP-b with default reaction to losses results more fair, as the temporary oscillations below the saturation point, caused by rate-halving, help starved flows to regain the access to the medium.

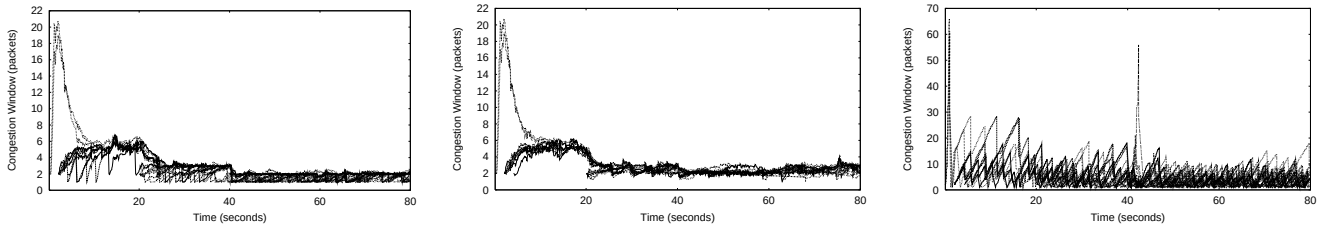
In terms of end-to-end delay it is clear from Fig. 7 that XCP-b outperforms TCP. The gains of XCP-b come mostly from reducing queuing delay in the path as XCP-b maintains small queues. While TCP, with queue sizes of 50 *packet* , adds an average of ≈ 180 ms to the end-to-end delay, XCP-b adds only ≈ 50 ms, including the transmission times across the several wireless hops.

In terms of throughput stability, we compare the average standard deviation σ of the throughput of the flows. The values are normalized by the average flow throughput μ. We consider samples of the throughput of each flow averaged during 1 second for a period of 30 seconds. The results are shown in Table 2. In average, XCP-b achieves a throughput whose normalized standard deviation is 45% of the Newreno's throughput normalized standard deviation. This behaviour is also observable in Fig. 8, where the evolution of the packet sequence number at the source of the flow is shown. The evolution in the case of XCP-b is much more smoother than in the case of TCP. Another fact observable from Table 2, is that XCP-b with no reaction to losses achieves a more stable throughput than XCP-b with reaction to losses. This was expected since XCP-b halves the rate when packets are lost, which causes oscillation in the throughput. We have also compared the values obtained from the simulation with a mesh access network to those ob-



(a) XCP-b (b) XCP-b with no reaction to losses (c) TCP Newreno

Figure 8: The evolution of the packet sequence number of the sources. It shows the total number of packets that have reached the destination.



(a) XCP-b (b) XCP-b with no reaction to losses (c) TCP Newreno

Figure 9: The congestion window variable of the sources.

Table 2: The average normalized standard deviation of the throughput of the flows, the total number of packets transmitted, and the collision probability at MAC level.

	$\frac{\sigma}{\mu}$ (%)	$\frac{\sigma}{\mu}$ (%) @1-hop	data pkts	$\frac{collisions}{total\_MAC\_pkts}$
XCP-b	22.3	4	20113	0.4285
XCP-b no loss	18	4	20620	0.44
Newreno	54	43	21063	0.436

tained when the flows destinations are stations that are only 1-hop away from the BS. The results show that for the case of the wireless multi-hop mesh network the throughput oscillation increases significantly. This supports the idea that in multi-hop wireless the access to the medium can be burstier and not as fair as in single hop wireless networks.

In terms of total network throughput, Table 2 shows the total number of unique data packets sent by the sources as well as the ratio between the number of MAC collisions and the total number MAC packets sent. The results show that XCP-b achieves a slightly lower throughput than TCP Newreno, even if it reduces MAC collisions. TCP Newreno is able to send about 2% more data packets than XCP-b with no reaction to losses, and 5% more than XCP-b with reaction to losses. The ratio between MAC collisions and MAC packets is approximately the same for TCP, and XCP-b with no reaction to losses, and slightly lower for XCP-b with reaction to losses. This occurs for a mixture of reasons. One is the TCP ACK-loss phenomenon which has the practical effect of diminishing the number transport layer ACK packets transmitted. This phenomenon is explained in more detail in [3]. Other reason is that XCP-b can drive the network below saturation for short periods, because it maintains small queues. TCP, on the other hand, has mid-to-full queues most of the time, which maintain the network at the saturation point at all times.

5.4 XCP-b parameter sensitivity

In the previous section we have studied the performance of XCP-b using the default parameter values defined in [3]. In this section we study how the variation of these parameter values affects the overall performance in terms of delay, total throughput and stability. XCP-b defines 3 parameters: κ, χ, and τ. κ represents the length at which the XCP-b queue will try to stabilize. χ controls the amount of bandwidth that is distributed by the XCP-b router when a fixed feedback is given. τ controls the threshold value of the exponential average of the queue, at which XCP-b switches from the fixed-feedback to a feedback based on queue variations. The original proposal of XCP-b suggests $\chi = 0.228$, $\tau = 0.178$, and $\kappa = 3$ *packet*. The value of χ is set so that the queue never overflows. The value of τ is set so that the XCP-b controller allows a few control intervals during which the queue is empty, before actually switching to the fixed-feedback. The number of intervals that the controller waits before making the switch in the feedback strategy, is a function of the period of oscillation of the queue response to capacity estimation errors. κ is chosen through simulation, where the value of 3 *packet* has proved to be sufficient in single hop 802.11 networks.

We now discuss the sensitivity of the performance of XCP-b to each of these 3 parameters. We evaluate the sensitivity through simulation, where we analyze the total number of

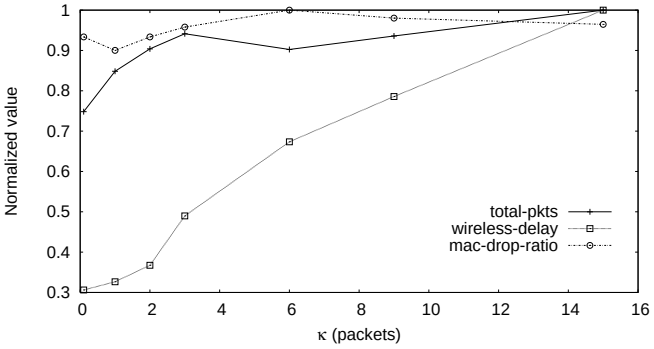


Figure 10: The average delay in the wireless path, the number of data packets sent, and the collision probability for different values of the parameter κ. All values are normalized to the highest value obtained.

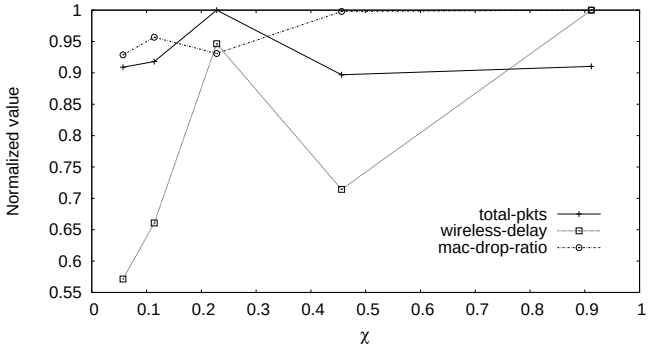


Figure 11: The average delay in the wireless path, the number of data packets sent, and the collision probability for different values of the parameter χ. All values are normalized to the highest value obtained.

data packets sent by sources, the ratio between MAC collisions and total MAC packets, and the delay in the wireless network. The simulation scenario is the same as in Fig. 5, and the simulation is setup as described in section 5.3.

5.4.1 sensitivity to κ

κ defines the length at which the queue will try to stabilize. Oscillations of the queue length around this value will be used to infer the amount of spare bandwidth of the channel. The higher the value of κ the more queuing delay will be introduced to the path. Also, higher temporary capacity oscillations will be supported without completely draining the queue. This contributes to the XCP-b controllers making less switches between the fixed-feedback and the feedback calculated using queue variations, which in the end contributes to a more stable queue. We ran simulations with $\kappa \in \{0.1, 1, 2, 3, 6, 9, 15\}$ and the results are shown in Fig. 10. As expected, the delay in the wireless part of network grows with κ. It is also possible to observe that the number of data packets sent by the sources grows fast until $\kappa = 3$ *packet*, after which the gain in terms of packets sent is modest. The number of MAC collisions also grows with the increase of κ, meaning that this increase drives the network nearer to the capacity saturation point. Overall, for small values of κ ($\kappa < 6$), the increase in utilization in considerable, until the network reaches a point near saturation, after which delay keeps rising linearly, with only a modest gain in terms of throughput.

5.4.2 sensivity to χ

χ controls the portion of bandwidth that is distributed when the medium is believed to be in under-utilization, which leads to empty queues, making it impossible to infer spare bandwidth from queue variations. Choosing a lower χ will lead to a slower adaptation of the used capacity to the full link capacity. Higher values of χ allow a faster adaptation to the medium capacity, but they will also produce higher spikes in the queue length as the increase step is larger. If $\chi > 0.228$, then queue overflow may occur. The result is that higher values of χ allow longer bursts. We ran simulations with $\chi \in \{0.057, 0.114, 0.228, 0.456, 0.912\}$. The results are presented in Fig. 11 and show that the increase of χ only increases the total number of data packets sent until $\chi = 0.228$. Further increase of χ results in the increase of the burstiness of the sources, which causes an higher MAC drop probability, resulting in less data packets reaching its destination successfully.

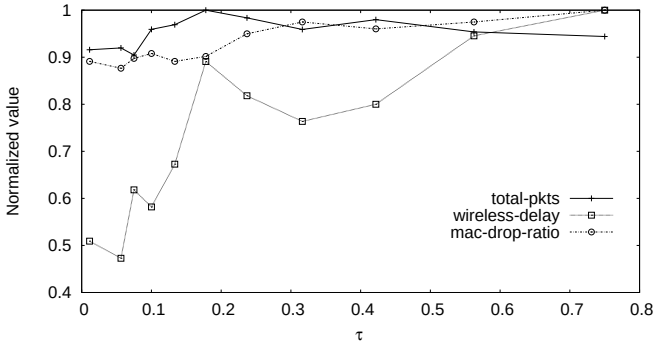


Figure 12: The average delay in the wireless path, the number of data packets sent, and the collision probability for different values of the parameter τ. All values are normalized to the highest value obtained.

5.4.3 sensitivity to τ

τ is the parameter that defines the number of control intervals the XCP-b controller allows the queue to be empty, before switching to the fixed-feedback strategy. The goal of this parameter is to minimize the number of switches between the two strategies of feedback: fixed-feedback, and feedback based on queue variations. The lower the value of τ the longer the periods of under-utilization will be. On the other hand, the number of switches between the two feedback strategies increases with the increase of τ, meaning that higher values of τ imply more oscillations. We ran simulations with $\tau \in \{0.013,\ 0.056,\ 0.075,\ 0.1001,\ 0.133,\ 0.178,\ 0.237,\ 0.316,\ 0.421,\ 0.563,\ 0.75\}$, and the results are shown in Fig. 12. The higher the value of τ the shorter the period during which the controller allows empty queues without switching to the fixed-feedback strategy. The results show that for low values of τ the total number of data packets sent is lower, while from a certain point ($\tau > 0.178$), the

increase in τ does not cause an increase in the total number of packets sent. As expected, for $\tau < 0.178$ the decreased utilization of the network is traduced in a lower delay in the wireless path, and lower collision probability.

6. CONCLUSIONS AND FUTURE WORK

In this paper we extend the validation of XCP-b to multi-hop wireless network scenarios. We evaluate the performance of XCP-b in simple chain-topologies, as well as in scenarios with mesh wireless access networks. The results show that XCP-b is viable in such scenarios, and that it outperforms regular TCP in terms of queuing delay, fairness and throughput stability. Such characteristics benefit primarily interactive and streaming applications which perform better with low latency and stable throughput. The results, however, show that throughput stability is much worser than in the case of single-hop wireless networks, mostly because in multi-hop 802.11 networks the hidden node problem may allow long bursts from single stations, and long idle periods for other stations. Another interesting aspect of XCP-b is that, unlike TCP, it maintains high efficiency when the capacity grows, making it suitable for next-generation 802.11n networks which claim to achieve speeds in the order of hundreds of Mbit/s.

We also study the sensitivity of the algorithm to the variation of its parameter values.ăWe conclude that the values suggested in [3] achieve the best results, but some trade-offs exist. Namely the increase of the stabilizing queue length κ can increase total utilization at the expense of extra queuing delay. On the other hand, the value of τ can be lowered to reduce this queuing delay.

The experiments also show that the congestion windows produced by XCP-b are closer to the theoretical limit found in [5], than the congestion window set by TCP Newreno. However, they are still above this limit as a result of queuing delay, which comes from the necessity of XCP-b build-up queues in the nodes. This increases competition for the medium, and produces sub-optimal spatial reuse.

We see as future work the incorporation of metrics, other than persistent queue variation into the feedback function of XCP-b. Such metrics could include the number of MAC collisions or MAC idle time, and they may remove the need to have queue build-up, removing the need to have the network in the saturation point. This approach may help to optimize spatial reuse in multi-hop wireless scenarios.

7. REFERENCES

[1] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[2] Simulator and scripts for "A Simulation Study of XCP-b in Multi-Hop Wireless Networks". http://pong.inescporto.pt/~fabrantes/code/.

[3] F. Abrantes and M. Ricardo. XCP for Shared-Acess Multi-Rate Media. *ACM Computer Communication Review*, 36:27–38, 2006.

[4] D. Allen. Hidden terminal problems in wireless lan's. IEEE 802.11 Working Group Papers, 1993, 1993.

[5] K. Chen, Y. Xue, and K. Nahrstedt. On Setting TCP Congestion Window Limit in Mobile Ad Hoc Networks. In *Proc. of IEEE ICC*, 2003.

[6] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt. Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks. *Elsevier Computer Commnications Journal*, 27:923–934, 2004.

[7] A. Falk and D. Katabi. Specification for the explicit control protocol (XCP). Internet Draft (draft-falk-xcp-spec-01), work in progress, October 2004.

[8] 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 *Proc. of the IEEE INFOCOM*, 2003.

[9] M. Gerla, K. Tang, and R. Bagrodia. TCP Performance in Wireless Multihop Networks. In *Proc.of the IEEE WMCSA*, 1999.

[10] V. Jacobson. Congestion avoidance and control. In *Proc. of ACM SIGCOMM*, 1988.

[11] R. Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling.* John Wiley and Sons, Inc., 1991.

[12] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proc. of ACM SIGCOMM*, 2002.

[13] V. Kawadia and P. R. Kumar. Experimental Investigations into TCP Performance over Wireless Multihop Networks. In *Proc. of the ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, 2005.

[14] J. Li, C. Blake, D.S.J.DeCouto, H. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *Proc. of the ACM/IEEE MOBICOM*, 2001.

[15] C. Perkins. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. of the ACM SIGCOMM*, 1994.

[16] J. Postel. Transmission control protocol. RFC 793, September 1981.

[17] Y. Su and T. Gross. WXCP: Explicit congestion control for wireless multi-hop networks. In *Proc. of IEEE IWQoS*, June 2005.

[18] Y. Zhang and M. Ahmed. A control theoretic analysis of XCP. In *Proc. of IEEE GLOBECOM*, March 2005.