XCP-Winf and RCP-Winf: Congestion Control Techniques For Wireless Mesh Networks

Luís Barreto[1], Susana Sargento[2]
[1]Instituto Politécnico de Viana do Castelo, Portugal
lbarreto@esce.ipvc.pt
[2] Instituto de Telecomunicações, Universidade de Aveiro, Portugal
susana@ua.pt

***Abstract*—In wireless networks a packet can be lost due to numerous reasons, such as congestion, medium related errors, routing and mobility. In these scenarios, the real congestion status of the network is crucial to develop accurate and efficient congestion control protocols. Some of the most known and recent protocols developed to provide faster and lighter congestion control are the eXplicit Control Protocol (XCP) and the Rate Control Protocol (RCP). These protocols have been proposed essentially to work in wired networks and environments; however, there are already new versions of XCP for wireless networks. Moreover, although these protocols estimate the available bandwidth of the links, this estimation is not accurate for wireless networks. This paper proposes new flow control protocols for wireless mesh networks, based on XCP and RCP, which we have designated as XCP-Winf and RCP-Winf. These congestion control mechanisms are supported on a new method to estimate the available bandwidth and the path capacity over a wireless network path, denoted as rt-Winf, through a cross-layer approach. The estimation is performed in real-time and without the need to intrusively inject packets in the network. This is accomplished by resorting to the CSMA-CA scheme with RTS/CTS packets to determine each node's channel allocation. The results obtained, through the simulation of these protocols in different scenarios, show that rt-Winf is able to significantly increase the efficiency of congestion control mechanisms.**

***Index Terms*—congestion control, available bandwidth, path capacity, measurements, performance, wireless networks.**

I. INTRODUCTION

In wireless networks packet loss is typically due to wireless channel impairments causing bit errors, handoffs due to mobility and, of course, possibly congestion. The most used congestion control protocol, Transmission Control Protocol (TCP) [1], assumes that a packet loss is always due to congestion in the network and, but not as often, of packet reordering. TCP does not respond well to packet losses due to bit errors and handoffs, making TCP-based applications suffering of poor performance. The eXplicit Control Protocol (XCP) [2] and the Rate Control Protocol (RCP) [3] are two of the most recent congestion control protocols. They rely on network interaction for congestion control improvement, since they need intermediate nodes, such as routers, to work and interact to support the congestion control. In wired networks they increase efficiency in the congestion control. However, as studied in [4], their performance in wireless networks, and more specifically in wireless mesh networks (WMNs), is decreased (their performance is even worse than TCP), since they are not able to accurately measure the available bandwidth of the wireless links.

As stated in [5], the limited capacity, and consequently available bandwidth, in WMNs, continue to be major factors that limit their performance. Therefore, severe congestion collapses are significant within WMNs. A congestion control scheme which provides an efficient and accurate sharing of the underlying network capacity among multiple competing applications is crucial to the efficiency and stability of WMNs. Then, it is of major importance to obtain accurately the link capacity and available bandwidth, and use these parameters actively in WMNs congestion control. Link capacity can vary due to a variety of factors, such as handoffs, channel allocation and, of course, channel quality. Therefore, it is of special interest to achieve an accurate monitoring of link capacity and available bandwidth, and use that information to congestion control and monitoring.

Estimation of link capacity has been widely studied, and can be achieved through either active or passive measurement [6]. Active measurement works by injecting measurement probe packets into the network, while passive measurement tools use existing data transmission. Active measurement has some important drawbacks, such as adding excess overhead and not always maintaining end-to-end semantics; passive measurement can be less reliable as it cannot rely on all the data. In [7] it was proposed rt-Winf, a new available bandwidth mechanism that accurately and passively measures the capacity and available bandwidth in WMNs.

This paper proposes the integration of rt-Winf in both XCP and RCP through a cross-layer approach, and defines two new congestion control protocols, XCP-Winf and RCP-Winf. These protocols use the capacity and available bandwidth of rt-Winf in their congestion control techniques to accurately evaluate the congestion status of the WMN links, and then take proper actions to avoid and reduce congestion. This integration will enable a significant improvement on performance as compared to base XCP and RCP, as it is shown through the obtained simulation results. This represents a significant step on the knowledge of WMNs behavior and congestion control, since it shows how cross-layer approaches can improve the network performance.

The rest of this paper is organized as follows. Section II briefly presents the related work. Then, section III describes the rt-Winf algorithm. In section IV, it is presented how rt-

978-1-61284-231-8/11/$26.00 ©2011 IEEE

Winf was integrated with XCP and RCP. Section V describes and discusses the results obtained through simulation, and finally, section VI presents the conclusions and future work.

II. BACKGROUND AND RELATED WORK

A. Capacity and Available Bandwidth Estimation

Link capacity estimation has been widely studied in wired networks. AbGet [8], PathChirp [9], IPerf [10], Pathload [11], IGI/PTR [6], Pathchar [12], CapProbe [13] are some examples. There are also developments with respect to wireless networks, such as AdHoc Probe [14], WBest [15] and IdleGap [16]. AdHoc Probe provides only the path capacity of the wireless channel. WBest calculates both capacity and available bandwidth.

IdleGap is a recent mechanism that obtains available bandwidth in wireless networks. IdleGap is focused on the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) scheme of wireless networks. It takes Network Allocation Vector (NAV) [17] into consideration, that is then used by the idle nodes which are waiting to transmit. It uses an approach to characterize the busy time and the total elapsed time, obtaining an *Idle Rate*. However, IdleGap uses the predefined IEEE802.11 header *DataRate* [18] value, which is not practical and real, thus leading to inaccurate and over-dimensioned estimation values.

The authors of IdleGap propose the consideration of 3 different states for a wireless node: *Sender*, *Receiver* and *Onlooker*. These states are distinguished on the *Idle Module*, which is the module used to determine the *Idle Rate*. The introduction of the *Idle Module* has an important disadvantage, that is the modification of the OSI Model [19], by the introduction of a new sublayer. rt-Winf algorithm will use some of the concepts of IdleGap, but it will not change the OSI model.

B. Congestion Control

The Transmission Control Protocol (TCP) [1] is the most used congestion control protocol in computer networks. TCP is known to have some limitations: unstable throughput, increased queuing delay, limited fairness. TCP assumes that the probability of a packet loss is higher than the one of a corrupted packet [20]. It is important to notice that this is not a true statement in WMNs.

Some new and specific congestion control mechanisms try to enhance TCP behavior in WMNs. Mechanisms like TCP-F [21], TCP-ELFN [22], TCP-BuS [23], ATCP [24] represent some examples of protocols for wireless networks in general. They concentrate on improving TCP's throughput by freezing TCP's congestion control algorithm during link-failure induced losses, especially when route changes occur. Those individual pieces of work differ in the manner of which these losses are identified and notified to the sender, and in the specific details of freezing TCP's congestion control algorithm. TCP-ELFN [23] explicitly notifies the TCP sender of routing failure causing the sender to enter standby mode. The sender re-enters the normal TCP mode on route restoration, identified by using periodic probe messages. In ATP [25], a flow receives the maximum of the weighted average of the sum of the queuing and transmission delay at any node traversed by the flow. ATP uses the inverse of this delay as the sending rate of a sender.

XCP [2] was designed to extract congestion information directly from routers. According to [26], ''XCP achieves fairness, maximum link utilization and efficient use of bandwidth''. XCP is also scalable, as per-flow congestion state is carried in packets. However, XCP requires changes to be made on all routers and end-systems in the network. A XCP network is composed of XCP sender hosts, receiver hosts and intermediate nodes, where queuing from the sender to the receiver occurs. XCP uses a feedback mechanism to inform the sender about the network conditions, that is, the maximum throughput. This feedback is accomplished by the use of a congestion header in each packet sent. Along the path, intermediate nodes update the congestion header. When the packet reaches the receiver, it copies the network information, obtained from the last intermediate router, into outbound packets of the same flow (normally acknowledgement packets).

The Rate Control Protocol (RCP) [3] is part of the 100x100 clean slate project [27]. RCP, similarly to XCP, is a congestion control algorithm. RCP was designed having in mind typical flows of typical users in today's Internet (traffic bursts). RCP uses the same feedback principle of XCP and tries to emulate processor sharing. However, it uses a different approach. Routers along the path do not determine incremental changes to the end-system's throughput, but determine the available capacity and the rate at which the end-system should operate.

It is shown in [4] that both XCP and RCP have poor performance when applied to WMNs, since they are not able to accurately measure the shared and multi-hop available bandwidth.

III. RT-WINF

The rt-Winf [7] mechanism was developed inspired by IdleGap [16], with the purpose to mitigate it's previously mentioned problems, being compatible with all systems and allowing to determine both the link capacity and the available bandwidth without overloading the network. rt-Winf does not introduce any changes to the OSI Model, as opposed to IdleGap, being able to obtain the necessary timing information to calculate the path capacity and the available bandwidth. Another important aspect of rt-Winf, relatively to IdleGap [16], is the effective calculation of the capacity, instead of using the *DataRate* value of the IEEE802.11 header [18]. rt-Winf can rely on the Request To Send (RTS) / Clear To Send (CTS) handshake or on probe packets. rt-Winf, as IdleGap, considers three different states for a node: *Receiver*, *Onlooker* and *Sender* states. rt-Winf proposes, then, three different methods to determine the capacity and available bandwidth. Figure 1 shows the different approaches for each state, while Figure 2 represents the state diagram of the rt-Winf tool. It is possible to observe each state's transitions.

If RTS/CTS packets are not present, rt-Winf can use probe packets in order to retrieve the transfer time values. Probe packets must be User Datagram Protocol (UDP) generated packets with altered Frame Control IEEE 802.11 header: Type Data and Subtype Reserved. It uses packets with *Frame Control Type* set to 10 (Data) and *Subtype* to 1001 (Reserved). This

State	Available Bandwidth	Capacity
On-looking	Captured RTS Packet? YES: $\mathrm{AB}=\mathrm{C}(1-\frac{\sum \mathrm{NAV}_{\mathrm{RTS}}}{\text{Total elapsed time}})$ NO: $\mathrm{AB}=\mathrm{C}(1-\frac{\sum \mathrm{NAV}_{\mathrm{CTS}}}{\text{Total elapsed time}})$	$\mathrm{C}=\frac{\text{Packet Size}}{ACK_{Time}-CTS_{Time}-2\mathrm{SIFS}}$
Sender	$AB=C_{Sender}(1-\frac{\sum \mathrm{NAV}_{\mathrm{CTS}}}{\text{Total elapsed time}})$	$C_{Sender}=\frac{\text{Packet Size}}{ACK_{Time}-CTS_{Time}-2\,SIFS}$
Receiver	$AB=C_{Receiver}(1-\frac{\sum \mathrm{NAV}_{\mathrm{RTS}}}{\text{Total elapsed time}})$	$C_{Receiver}=\frac{\text{Packet Size}}{DATA_{Time}-RTS_{Time}-3\mathrm{SIFS}}$

Figure 1. rt-Winf Algorithm.

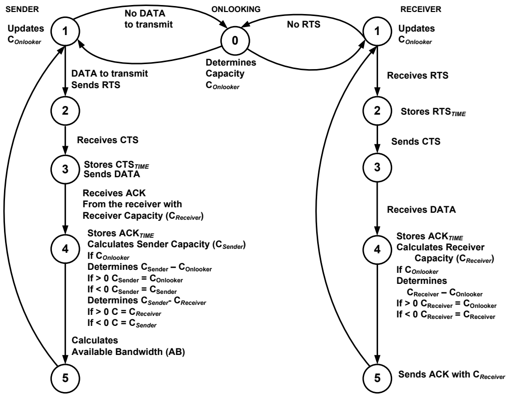

Figure 2. rt-Winf Sender, Receiver and Onlooking State Diagrams.

way the *Sender* and the *Receiver* can successfully differentiate these packets from the ordinary data packets. IEEE802.11 standard defines that, for each successfully received packet, it must be sent a MAC ACK packet [18]. The whole process is very similar to the one with the RTS/CTS handshake. The generated packets are used to retrieve the capacity and available bandwidth values, according to Equation 1 and Equation 2. To be fully operational, both *Sender* and *Receiver* must be running the rt-Winf mechanism.

$$C = \frac{PacketSize}{TransferTime} \tag{1}$$

where *TransferTime* is equal to $ACKTime - DataTime$.

$$AB = 1 - \left(\frac{\sum TransferTime}{TotalElapsedTime}\right) \times C \tag{2}$$

In a common VoIP call using G.711 codec [28], the overhead introduced by this mechanism is $\sim 1.66\%$. For a flow with more than 1Mbps, the overhead is less than $\sim 0.15\%$.

IV. XCP-WINF AND RCP-WINF

XCP-Winf and RCP-Winf rely on the main functioning principles of XCP and RCP, but use the information provided by rt-Winf to determine the shared and multi-hop available bandwidth in the WMN links. As rt-Winf obtains available bandwidth and capacity values in the MAC layer, this information has to be accessed by XCP and RCP. The rt-Winf information is sent to the network layer through a cross layer communication process. For this communication system, it was used a shared database architecture, with a set of methods to get/insert information from/in a database accessible by all protocol layers. One example of such architecture is the MobileMan cross-layered network stack [29]. A generic XCP-Winf/RCP-Winf system is represented in Figure 3. After obtaining the available bandwidth and the link capacity, rt-Winf inserts that information in the shared database and, then, XCP and RCP access that information and fine-tune their functions with the accessed information. One of the main advantages of using MobileMan is its reduced overhead, as stated on its architecture reference. Another advantage is its low level of complexity, as it takes full advantage of what is already offered by the network and rt-Winf. Since co-operation between the different protocols, MAC and XCP/RCP, takes place in the database the normal stack operation is not compromised. Whenever rt-Winf collects information, it will publish this to the repository and thus making it available for every other protocol. This is done node by node, being XCP/RCP main functions responsible for the interaction between nodes.

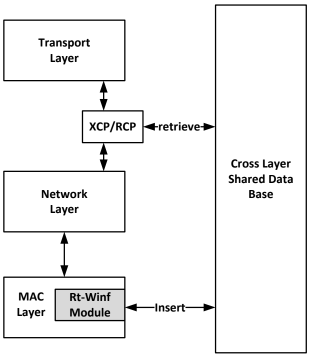

Figure 3. XCP-Winf/RCP-Winf System.

A. XCP-Winf and RCP-Winf Functions

This section briefly describes some of the XCP-Winf and RCP-Winf functions. Compared to base XCP/RCP, the functions that are changed are the Sender and Router functions. The XCP/RCP Sender uses the *Sender* state of the rt-Winf algorithm, and the XCP/RCP Router uses the *Onlooker* state. Next, we present the corresponding algorithms for the XCP-Winf Sender and Router functions. The XCP-Winf Receiver is just responsible for copying the $Delta\_Throughput$ data that it arrives into the $Reverse\_Feedback$ field of outgoing packets. A XCP-Winf Receiver operates in a similar way as a XCP Receiver. When acknowledging a packet, the XCP-Winf Receiver copies the congestion header from the data packet to the corresponding acknowledgment packet, and acknowledges the data packet in the same way as a TCP receiver.

When operating as a XCP-Winf Sender, several calculations need to be performed. The pseudo-code of a XCP-Winf Sender is presented in Algorithm 1. The XCP-Winf operations are basically the same as the standard XCP, except that it uses rt-Winf to obtain the link capacity and available bandwidth and, then, it obtains the $Delta\_Throughput$. If no additional capacity is needed, the $Desired\_Troughput$ will be equal to zero, and the packet will be immediately sent. If the value of $Delta\_Throughput$ exceeds $Available\_Bandwidth_{Winf}$, it is reduced to the current value of $Available\_Bandwidth_{Winf}$.

The *Onlooker* operations for a XCP-Winf Router system are divided in four moments. Those moments are: when a packet

Algorithm 1: XCP-Winf Sender Algorithm.

```
/* Available Bandwidth and Capacity Estimation */
$Desired\_Throughput$: senders desired change in throughput.
$Available\_Bandwidth_{Winf}$ : rt-Winf obtained available bandwidth.
$Delta\_Throughput$: desired or allocated change, per packet, in throughput.
$C_{Winf}$: rt-Winf obtained Capacity.

Access Cross Layer Shared Database;
Retrieve Available Bandwidth and Capacity;
/* Obtain Delta Throughput */
Throughput = $C_{Winf}$;
$Desired\_Throughput <= Available\_Bandwidth_{Winf}$;
$Delta\_Throughput = \frac{Desired\_Throughput - C_{Winf} \times 1000}{C_{Winf} \times \frac{RTT}{MSS}}$;
/* Send a packet */
Update Congestion Header Delta Throughput Field;
Send Packet;
```

arrives, when a packet departs, the control interval timeout of a packet and the assessment of the persistent queue. Next, it is presented the pseudo-code for the Router/Onlooker Control Interval Timeout Operations (Algoritm 2). It is possible to observe that some of its important parameters are retrieved and managed by rt-Winf. The *Aggregated Feedback* is obtained using link capacity and available bandwidth obtained by the rt-Winf mechanism. The same principles prevail in the other XCP-Winf Router/Onlooker moments.

Moreover, the same principles are used in RCP-Winf, using the same operations of RCP, but obtaining rt-Winf available bandwidth and capacity values. For example, in a RCP-Winf Sender, the link capacity used is obtained by rt-Winf ($rcp\_bottleneck\_rate$ = rt-Winf link capacity), which is made available by the cross-layer communication mechanism. Algorithm 3, which is exclusive of the RCP implementation, shows some of the per-packet operations performed by a RCP-Winf router when the rate estimation timer expires. Comparing to the base RCP algorithm, the $link\_rate$ is, now, obtained using the capacity value of rt-Winf, i.e. C_{Winf}.

As a final remark, it is possible to observe that XCP-Winf and RCP-Winf implementations differ from the standard implementation in the way link capacity and available bandwidth are obtained and used. The process of cross-layer communication was developed to transfer the MAC-layer informations to the congestion control protocols.

V. SIMULATION RESULTS

This section shows the simulation results of the proposed congestion control approaches, XCP-Winf and RCP-Winf. The results are obtained using the ns-2 simulator [30]. In the simulations we used various mesh topologies scenarios. The mesh topologies defined were: a grid of 5, 9, 12 and 16 fixed mesh nodes, and an ad-hoc network. In all mesh topologies, it was used a combination of 3, 4, 5, 6 and 7 mobile nodes. The mobile nodes were, simultaneously, sources and sinks. The routing protocol used was the Destination-Sequence Distance-Vector (DSDV) [31]. All simulations last 300 seconds. The simulations were repeated 10 times with different ns-2 seed values, and both the mean and 95% confidence intervals are presented in the graphics below. The configured default transmission range was 250 meters, the default interference range is 500 meters, and the channel data rate is 11 Mbps. For the data transmissions, it was configured a File Transfer Protocol (FTP) application with packets of 1500 bytes and a Constant Bit Rate (CBR) application generating a 64 Kbps UDP traffic. In the mobility scenarios, the ns-2 *setdest* tool was used. This tool generates a random node movement pattern. *Setdest* was configured with a minimum speed of 10 m/s, a maximum speed of 30 m/s and a topology boundary of 1000x1000 meters. All results were obtained from ns-2 trace files, with the help of *trace2stats* scripts [32] adapted to our own needs.

Algorithm 2: XCP-Winf Router/Onlooker Control Interval Timeout Operations.

```
$avg\_rtt$: average rtt value, used to determine the control interval.
$F_{Winf}$: Aggregated Feedback, uses rt-Winf values.
$C_p$: positive feedback scale factor.
$C_n$: negative feedback scale factor.
$residue\_pos\_fbk$: pool of available positive capacity a router has to allocate.
$residue\_neg\_fbk$: pool of available negative capacity a router has to allocate.
$MIN\_INTERVAL$: propagation delay on link, value between 5 and 10 ms.

On estimation control timeout do:
$avg\_rtt = \frac{sum\_rtt\_by\_throughput}{sum\_inv\_throughput}$;
$input\_bw = Available\_Bandwidth_{Winf}$;
$F_{Winf} = a \times (C_{Winf} - input\_bw) - b \times \frac{queue}{avg\_rtt}$;
$shuffled\_traffic = max(0, 0.1 \times input\_bw - |F_{Winf}|$;
$residue\_pos\_fbk = shuffled\_traffic + max(F_{Winf}, 0)$;
$residue\_neg\_fbk = shuffled\_traffic + max(-F_{Winf}, 0)$;
$C_p = \frac{residue\_pos\_fbk}{sum\_inv\_throughput}$;
$C_n = \frac{residue\_neg\_fbk}{input\_traffic}$;
$input\_traffic = 0$;
$sum\_inv\_throughput = 0$;
$sum\_rtt\_by\_throughput = 0$;
$ctl\_interval = max(avg\_rtt, MIN\_INTERVAL)$;
$timer.reschedule(ctl\_interval)$;
```

Algorithm 3: RCP-Winf Router/Onlooker Rate Estimation Timer Operations.

```
$rcp\_rate$: the bandwidth offered to a flow.
$MIN\_RATE$: the minimum value for rcp_rate.
$ETA$: a constant value.
$C_{Winf}$ : rt-Winf obtained Capacity.

On rate estimation timer timeout do:
......
if ($rcp\_rate < MIN\_RATE$) then
    $rcp\_rate = MIN\_RATE$;
else if ($rcp\_rate > ETA \times C_{Winf}$) then
    ($rcp\_rate = ETA \times C_{Winf}$);
......
```

Next we present, analyze and compare the results of both versions of XCP and RCP, the based ones and the x-Winf versions. We also include TCP for comparison purposes. The results show throughput and the number of received packets. Although we do not present the delay results due to space limitations, we include the general conclusions on the delay impacts.

Figure 4 and Figure 5 show the previously referred performance metrics for five different cases. In each case, it was used a fixed number of 16 mesh nodes and a variable number, from 3 to 7, of mobile nodes. Figure 4 shows how throughput is improved in XCP and RCP with rt-Winf, and also, how

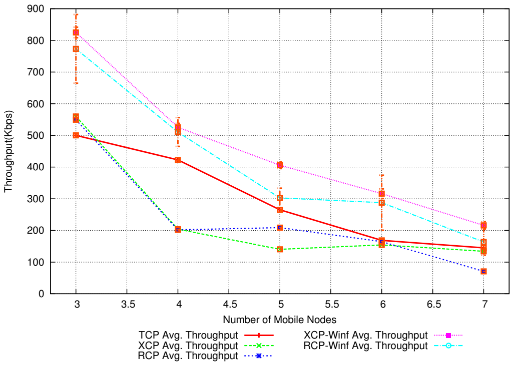

Figure 4. 16 Mesh Nodes - Variable Number of Mobile Nodes, Throughput.

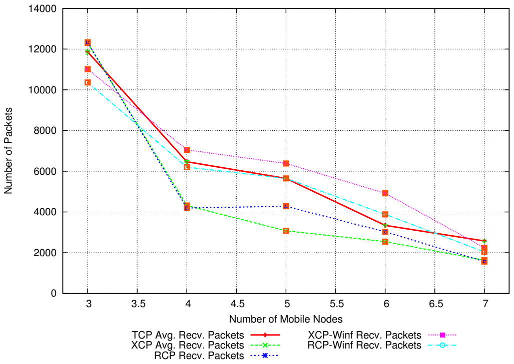

Figure 5. 16 Mesh Nodes - Variable Number of Mobile Nodes, Received Packets.

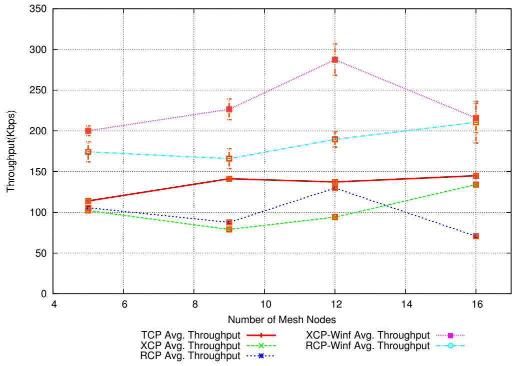

Figure 6. Variable Number of Mesh Nodes - 7 Mobile Nodes, Throughput.

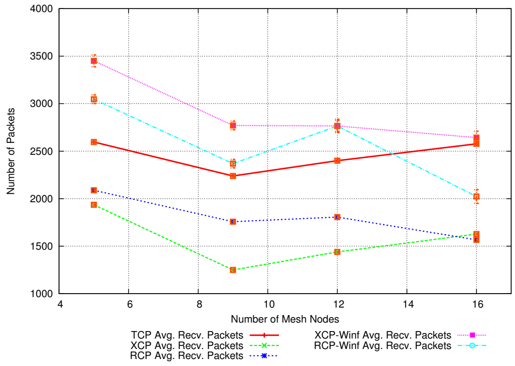

Figure 7. Variable Number of Mesh Nodes - 7 Mobile Nodes, Received Packets.

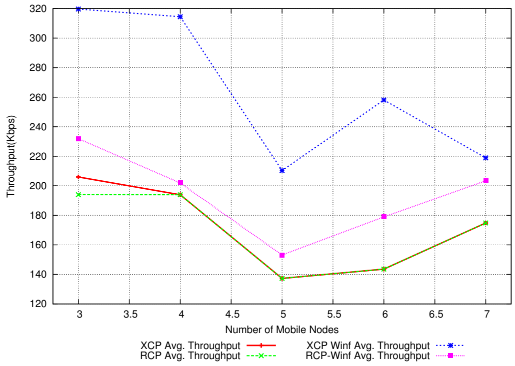

Figure 8. 16 Mesh Nodes - Variable Number of Mobile Nodes, CBR Throughput

it is improved compared to TCP. The throughput values of XCP-Winf are $\sim$ 47% to $\sim$ 60% better than the ones with TCP, while with the base XCP throughput values were worse than TCP. For RCP-Winf, the percentages when compared to TCP are between $\sim$ 17% and $\sim$ 56%. In terms of received packets, as observed in Figure 5, it is also possible to see that with rt-Winf integrated, both XCP and RCP can receive more packets, which reflects a lower rate of lost packets. This is due to the fact that XCP-Winf and RCP-Winf, with accurate link capacity and available bandwidth, are using more efficiently the medium and improving each node queue management. Then, the nodes, and of course the network, can transmit with a higher rate and less losses. As more packets are transmitted, more throughput is obtained and the medium is better used, it is possible to infer that both XCP-Winf and RCP-Winf are more stable and fair; in the same conditions, it is possible to send more information with a higher rate. The delay values, although not shown, are significantly decreased by one order of magnitude in the XCP/RCP-Winf versions, from around 1 sec to 100 msec in the 16 mesh nodes scenario.

Figure 6 and Figure 7 show the performance metrics, but for a fixed number of mobile nodes (7), and a variable number of mesh nodes (5, 9, 12 and 16 mesh nodes). As XCP and RCP need, to operate, that all nodes in the network exchange information, the number of collisions increases, leading to higher losses, and consequently lower number of packets received and lower throughput (although not shown, the delay is high). With the integration of rt-Winf, it is possible to observe that XCP-Winf and RCP-Winf have good throughput values and can transmit more packets, allowing a much better medium usage.

Finally, Figure 8 shows the obtained results with a CBR UDP application (simulating a VoIP application), for the 16 mesh nodes scenario and variable number of mobile nodes. It is possible to observe that, with rt-Winf integrated, the throughput results are considerably better, but, still, reflect the problems that XCP and RCP have in controlling congestion when the traffic is UDP. Once more, RCP-Winf reflects its base development for bursty traffic. The CBR application is sending data at a constant rate but, with more mobile nodes sending data, more collisions will occur and more bursts of traffic will be present in the network. This situation will allow RCP to react more precisely and more naturally, resulting, as the number of mobile nodes increases, in better throughput results.

The behavior of XCP-Winf and RP-Winf was also analyzed in ad hoc scenarios, but due to space limitations they are not presented here. Those scenarios also show that XCP-Winf and RCP-Winf improve XCP and RCP behavior in wireless ad hoc networks.

The results show that the integration of rt-Winf in XCP and RCP improves significantly their behavior. The available bandwidth and capacity evaluation of rt-Winf, and the cross-layer information, are important and surely make XCP and RCP behave more consistently and with better channel utilization, which also leads to less channel losses (more received packets). The use of rt-Winf in the the mesh nodes (onlooking state) makes the feedback mechanism more accurate, as all nodes in the network can determine available bandwidth and capacity, and send that information to the other nodes that are participating in the communication.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed a cross-layer approach to congestion control in WMNs. It presented two congestion control protocols, XCP-Winf and RCP-Winf, that integrate the measurements of WMN status through a new passive monitoring tool, rt-Winf. It measures the wireless capacity and the available bandwidth of WMN links, and feeds this information to RCP and XCP. rt-Winf uses information already available on the network: it can rely on the CTS/RTS/ACK messages handshake or on small probes. These packets provide time information, allowing to know each node's channel allocation.

The performance evaluation study of the proposed congestion control approaches shows that the rt-Winf algorithm improves significantly XCP and RCP behavior, making them more stable and fair. To obtain the available network capacity, both base XCP and RCP would need that all nodes in the network cooperate, which increases network overhead, specially when dealing with wireless mesh networks. Using rt-Winf, all this information comes from the MAC layer, where link capacity and available bandwidth calculations are performed without interfering in the network dynamics.

As future work, we plan to work on the wider evaluation of the congestion control approaches (with new scenarios and new comparison baselines and protocols). An effort will also be made in understanding how the throughput (goodput) improvement and packet transmission success is affected by different conditions and parameters.

REFERENCES

[1] J. Postel, "Transmission control protocol," RFC 793, Defense Advanced Research Projects Agency, September 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[2] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," *ACM SIGCOMM*, August 2002.

[3] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," *ACM SIGCOMM*, 2006.

[4] L. Barreto and S. Sargento, "TCP, XCP and RCP in wireless mesh networks: An evaluation study," in *15th IEEE Symposium on Computers and Communications (IEEE ISCC'10)*, Riccione, Italy, 6 2010.

[5] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445 – 487, 2005.

[6] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, August 2003.

[7] B. Res, L. Barreto, and S. Sargento, "rt-winf: Real time wireless inference mechanism," in *IEEE Globecom 2010 Workshop on Mobile Computing and Emerging Communication Networks (MCECN 2010)*, Miami, Florida, USA, 2010.

[8] D. Antoniades, A. Papadogiannakis, E. P. M., and C. D., "Available bandwidth measurement as simple as running wget," *Passive and Active Measurement (PAM) Workshop*, 2006.

[9] V. J. Ribeiro, R. H. Riedi, R. G. B., J.N., and L. C., "Pathchirp: Efficient available bandwidth estimation for network paths," *Passive and Active Measurement (PAM) Workshop*, April 2003.

[10] "IPerf." [Online]. Available: http://dast.nlanr.net/Projects/Iperf/

[11] J. Manish and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics and relation with TCP throughput," *ACM SIGCOMM*, August 2002.

[12] V. Jacobson, "Pathchar: A tool to infer characteristics of internet paths," *MSRI talk*, April 1997.

[13] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique." *ACM SIGCOMM*, vol. 34, no. 4, August 2004.

[14] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "Adhoc probe: Path capacity probing in wireless ad hoc networks," *Wireless Networks*, vol. 15, no. 1, pp. 111–126, 2009.

[15] M. Li, M. Claypool, and R. Kinicki, "Wbest: a bandwidth estimation tool for ieee 802.11 wireless networks," *IEEE LCN*, October 2008.

[16] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim, "Bandwidth estimation in wireless LANs for multimedia streaming services," *Advances in Multimedia*, vol. 2007, no. 1, pp. 9–9, 2007.

[17] M. S. Gast, *802.11 Wireless Networks - The Definitive Guide*, 4th ed. O'reilly, 2002.

[18] *IEEE Std 802.11 - IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society Standard, 2007. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[19] H. Zimmermann, "OSI reference model- the ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, April 1980.

[20] B. A. Fourouzan, *TCP/IP Protocol Suite*, 2nd ed. McGraw-Hill Higher Education, 2002.

[21] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback-based scheme for improving tcp performance in ad hoc wireless networks," 2001.

[22] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, US: ACM, 1999.

[23] D. Kim, T. C.-K., and Y. C., "Tcp-bus: improving tcp performance in wireless ad hoc networks," *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, vol. 3, pp. 1707 –1713 vol.3, 2000.

[24] J. Liu and S. Singh, "Atcp: Tcp for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1300 –1315, jul. 2001.

[25] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and A. R. Sivakumar, "Atp: a reliable transport protocol for ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 4, no. 6, pp. 588 – 603, nov. 2005.

[26] A. Falk and D. Katabi, *Specification for the Explicit Control Protocol (XCP), draft-falk-xcp-spec-03.txt*, Information Sciences Institute Internet-Draft. [Online]. Available: http://www.isi.edu/isixcp/docs/draft-falk-xcp-spec-00.html

[27] "100x100 clean state project." [Online]. Available: http://100x100network.org/

[28] "Recommendation ITU-T G.711.0," ITU, 2005. [Online]. Available: http://www.itu.int/rec/T-REC-G.711/

[29] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *Computer*, vol. 37, pp. 48–51, 2004.

[30] "The network simulator - ns-2," 2001. [Online]. Available: http://www.isi.edu/nsnam/ns/

[31] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM*, 1994.

[32] M. Fiore, "trace2stats." [Online]. Available: http://www.tlc-networks.polito.it/fiore/index.html