Cross-Layer Information to Enhance TCP-AP Congestion Control in Wireless Networks

Luís Barreto

Abstract

The dynamic and instability of wireless links make congestion control an important research subject in wireless networks. It is important to define strategies that accurately evaluate the characteristics of the wireless links and what can be used by congestion control protocols. It is known that TCP experiences serious performance degradation problems in wireless networks. It has been showed that new variants of TCP, defined fo wireless congestion control, such as TCP-AP, do not evaluate accurately the capacity and available link bandwidth in wireless networks. In this paper, we propose a new congestion control protocol for wireless networks, based in TCP-AP and in a new cross layer information (CLI). We name the proposed protocol TCP-AP with *CLI*. It relies on the MAC layer information gathered by a method that accurately estimates the available bandwidth and the path capacity over a wireless network path and introduces the concept of node count fair bandwidth share effect. The new congestion control mechanism is evaluated in different scenarios, in wireless mesh and ad-hoc networks, and compared against several approaches for wireless congestion control. It is shown that TCP-AP with *CLI* outperforms the base TCP-AP, showing its stable behavior and better channel utilization.

***Keywords*—Congestion control, available bandwidth, path capacity, measurements, performance, wireless networks.**

I. Introduction

Wireless networks have major factors that limit their performance, this factors as stated in [1] are their limited capacity and available bandwidth. This results in severe congestion collapses. 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 wireless networks. Actively using link capacity and available bandwidth for congestion control will surely make these networks more efficient. Link capacity can vary due to a variety of factors, such as , channel allocation and, of course, channel quality. [2] presents a new mechanism for measuring wireless link capacity and available bandwidth, called *rt-Winf*. *rt-Winf* uses the information already present in the network and available at the MAC layer. Another important characteristic of *rt-Winf* is that it can be used by any existing wireless equipment.

To address the congestion control problems of wireless networks, new congestion control techniques have been proposed. The Transmission Control Protocol with Adaptive Pacing (TCP-AP) [3] is a congestion control mechanism based on TCP [4], specifically designed for ad hoc multi-hop wireless networks, it uses a 4-hop propagation delay technique. TCP-AP uses a hybrid scheme between a pure rate-based transmission control and TCP's use of the congestion window. However, TCP-AP as studied in [5] is very conservative and is not using efficiently the medium. TCP-AP relies only on the 4-hop propagation delay to evaluate link available bandwidth and capacity, thus, not taking into consideration all the factors that influence link evaluation.

New simulation results, presented in this paper, conducted in wireless ad hoc scenarios clear state that TCP-AP lacks of efficiency and is not using correctly the medium, thus, not evaluating correctly the parameters that are real constraints in such networks. We propose, for improving TCP-AP behavior and having in mind the previous considerations, the integration of a new on-line capacity and available bandwidth estimation technique, called *rt-Winf*, with TCP-AP through a cross layer communication process. Simulation results show that the *rt-Winf* integration is improving TCP-AP performance. However, TCP-AP with *rt-Winf* still reflects some of TCP-AP flaws, especially concerning fairness and not using the entire network information, as relies for situations over the 4-hop propagation delay mainly the TCP Additive Increase Multiplicative Decrease (AIMD) mechanism. Thus, it is also important to improve its operations with the knowledge of all nodes along the path contending for available bandwidth and capacity, introducing the fairness factor and the network interaction behavior. New simulation results show that the consideration of the node path effect and the integration of *rt-Winf* clear improve base TCP-AP performance. The simulation results were conducted on both ad hoc and mesh wireless networks. These considerations represent in terms of wireless networks congestion control and behavior a significant step towards their knowledge.

The remaining of this paper is organized as follows. Next section, section II, briefly presents the related work on congestion control mechanisms for wireless networks. Then, section III describes how *rt-Winf* is integrated with TCP-AP and how the node path contention count effect and collision probability are included to obtain the proposed TCP-AP with *CLI* protocol. Section IV describes and discusses the results obtained through simulation, using mesh and ad-hoc scenarios with different characteristics. Finally, section V presents the conclusions and future work.

II. Related Work

New efforts have been made to improve congestion control in wireless networks. The Wireless Control Protocol (WCP)

Luís Barreto is with the Instituto Politécnico de Viana do Castelo, Portugal, and is a researcher in the Instituto de Telecomunicações, Universidade de Aveiro, Portugal (e-mail: lbarreto@esce.ipvc.pt)

[6], WCP with Capacity Estimation (WCPCap) [6], Cooperative Neighborhood Airtime-limiting (CNA) [7], HOP [8], EZ-Flow [9] and Neighborhood Random Early Detection (NRED) [10] are some examples.

TCP, as the most used congestion control protocol, has also been the underlying development for some congestion mechanisms in wireless environments, TCP-AP [3] is one example. More recent developments, like XCP-b [11], XCP-Winf [5] and RCP-Winf [5], are based on rate based congestion protocols like the eXtensible Control Protocol (XCP) [12] and the Rate Control Protocol (RCP) [13].

WCP is an AIMD-based rate-control protocol for multi-hop wireless networks. WCP was designed with the goal to be used on networks with arbitrary traffic pattern. During congestion, WCP signals all flows in a neighborhood of congestion and sets the control interval to the maximum Round Trip Time (RTT) of any flow in the neighborhood. WCP explicitly exchanges congestion information within a neighborhood, and all nodes within the neighborhood mark packets with congestion indicators, triggering rate reductions at the source.

WCPCap is a distributed rate controller that estimates the available capacity within each neighborhood, and divides this capacity to contending flows. With WCPCap it is evident that considering wireless congestion collectively over a neighborhood of a link is essential to any future design of wireless congestion control. WCPCap uses a sophisticated stochastic model for estimating the achievable rate region, given packet loss rates, topology, and flow information. It then allocates the achievable capacity fairly across flows, sending feedback to the sources.

CNA is a hybrid approach, in that it explicitly allocates the channel resources, but provides only imprecise feedback to the source. CNA achieves efficient airtime allocation by distributing available airtime within a wireless neighborhood, then monitoring the air utilization and dynamically redistributing unused airtime to improve overall airtime usage. The authors of CNA claim that it achieves transparency, low overhead, and responsiveness. CNA considers airtime to be the fraction of the time that a wireless link can occupy the shared channel; it does not consider the time a node is waiting to transmit.

HOP is a clean-slate design of hop-by-hop congestion control. HOP tries to use reliable per-hop block transfer as a building block. HOP is referred by its authors as: fast, because it eliminates many sources of overhead as well as noisy end-to-end rate control; robust to partitions and route changes, because of hop-by-hop control as well as in-network caching, and simple, because it obviates complex end-to-end rate control as well as complex interactions between the transport and link layers.

EZ-Flow is a back-pressure congestion control mechanism which does not require explicit signaling. A back-pressure mechanisms flow control allows loss-free transmission by having gateways verify that the next gateway has sufficient buffer space available before sending data, thus EZ-Flow is a cooperative congestion control. EZ-flow operates by adapting the minimum congestion window parameter at each relay node, based on an estimation of the buffer occupancy at its successor node in the mesh.

NRED identifies a subset of flows which share channel capacity with flows passing through a congested node. But, it identifies only a subset of contending flows: it misses flows that traverse two hop neighbors of a node without traversing its one hop neighbors. Moreover, the mechanism to regulate the traffic rates on these flows is quite a bit complex (it involves estimating a neighborhood queue size and using RED-style marking on packets in this queue). NRED has an important disadvantage, being intimately tied to a particular queue management technique (RED) and requires special hardware for channel monitoring.

TCP-AP uses a 4-hop propagation delay technique, it considers a hybrid scheme between a pure rate-based transmission control and TCP's use of the congestion window to trigger new data packets to be sent into the network. A TCP sender adaptively sets its transmission rate using an estimate of the current 4-hop propagation delay and the coefficient of variation of recently measured round-trip times. The 4-hop propagation delay describes the time elapsed between transmitting a TCP packet by the TCP source node and receiving the packet at the node which lies 4 hops apart from the source node along the path to the destination.

XCP-b is a XCP based congestion control mechanism, it tries to extend XCP for shared-access, multi-rate wireless networks by calculating, using very complex heuristics, the available bandwidth of the wireless channel. XCP-b uses indirect parameters such as queue sizes and number of link layer retransmissions to obtain the desired measurements. XCP-b major drawback is that it becomes inefficient over highly dynamic wireless networks. In wireless environments with few nodes and less mobility, XCP-b can obtain good performance results in terms of stability, fairness, and convergence.

XCP-Winf and RCP-Winf are two new congestion control mechanisms, that use MAC layer information through a cross layer communication process. The *rt-Winf* algorithm performs link capacity and available bandwidth calculations without interfering in the network dynamics, and without increasing network overhead, these parameters are then passed to the congestion control mechanisms based on explicit congestion notifications, XCP and RCP, to accurately determine the network status and act accordingly. The evaluation results of XCP-Winf and RCP-Winf, obtained through ns2 [14] simulations, show that the rt-Winf algorithm improves significantly XCP and RCP behavior making them more efficient and stable.

Table I qualitatively compares the previous referred mechanisms along some dimensions.

III. TCP-AP WITH *CLI*

A. *TCP-AP with rt-Winf*

TCP-AP with *rt-Winf* information relies on the main functioning principles of TCP-AP, but uses information provided by *rt-Winf* [2] to determine the link capacity and available bandwidth. The base TCP-AP considers fot its capacity and available bandwidth estimations transport layer information (RTT values). Thus, this technique is not very accurate introducing errors on the the congestion control process. Using

	Cross-Layer	Contr. Param.	TCP based	AIMD based	Rate based	Inter. Node Feedback	MAC Multi Rate
NRED	Yes	Buffer				Yes	
XCP-Winf RCP-Winf	Yes	Rate			Yes	Yes	Yes
WCP		Rate		Yes		Yes	
WCPCap		Rate		Yes	Yes	Yes	
CNA		Window				Yes	Yes
EZ-Flow	Yes	Buffer				Yes	
HOP		Window					Yes
TCP-AP		Window	Yes	Yes	Yes		Yes
XCP-b	Yes	Buffer				Yes	

Table I
CONGESTION CONTROL MECHANISMS COMPARISON.

MAC layer information to update the available bandwidth and link capacity estimation, allows the congestion control mechanism to be more reliable and effective.

As *rt-Winf* determines link capacity and available bandwidth in the MAC layer this information has to be accessed bay TCP-AP through a cross layer communication process. For the cross layer communication process it was used the MobileMan [15] cross-layered network stack. This communication system uses a shared database architecture, with a set of methods to get/insert information from/in the database accessible by all protocol layers.

Compared to the base TCP-AP, TCP-AP with *rt-Winf* changes the way each node calculates the four hop delay (FHD) and the average packet queuing delay per node (t_q), with the *rt-Winf* link capacity and available bandwidth values. Thus,

$$t_q = \frac{1}{2}\left(\frac{T_{RTT}}{h} - \frac{S_{data} - S_{ack}}{C_{Winf}}\right) \quad (1)$$

where T_{RTT} represents the RTT value, h represents the number of hops between sender and receiver, S_{data} is the size of the data packet and S_{ack} the size of the ACK packet. Finally, C_{Winf} corresponds to the *rt-Winf* link capacity. The previous equation allows to update the 4-hop delay (FHD) by:

$$FHD = 4 \times \left(t_q + \frac{S_{data}}{AB_{Winf}}\right) \quad (2)$$

where AB_{Winf} is the *rt-Winf* available bandwidth.

As the standard TCP-AP, considers for its capacity and available bandwidth estimations transport layer information (RTT values), this technique is not very accurate introducing inefficiency to the congestion control process. With the update of the available bandwidth and link capacity estimation, using information provided by the MAC layer, the congestion control mechanism is more reliable and effective.

Considering that a high density and high mobility network suffers from a large number of collisions, *rt-Winf* mechanism was updated with the effect of collision probability. Notice that *rt-Winf* works on the IEEE 802.11 [16] MAC layer that uses the Distribution Coordination Function (DCF) as the access method. This function is based on the CSMA-CA principle in which a host wishing to transmit senses the channel, waits for a period of time and then transmits if the medium is still free. If the packet is correctly received, the receiving host sends an ACK frame after another fixed period of time. If the ACK frame is not received by the sending host, a collision is assumed to have occurred. Therefore, to improve efficiency and reliability of TCP-AP with *rt-Winf*, collision probability is accounted for. When a sender cannot transmit due to collision, the backoff mechanism is activated. This mechanism is also consuming bandwidth. This extra bandwidth, C_{extra}, is defined by:

$$C_{extra} = \frac{T_{DIFS}}{T_{backoff}} T_m \quad (3)$$

, where T_{DIFS} is the IEEE 802.11 DCF Interframe Space, $T_{backoff}$ is the medium backoff time and T_m is the time between the transmission of two packets. The collision probability (P_c) can then be defined as $1 - C_{extra}$. Applying this result to the rt-winf inference mechanism, the available bandwidth (AB) becomes:

$$AB = P_c \times AB_{Winf} \rightarrow AB = (1 - C_{extra}) \times AB_{Winf} \quad (4)$$

B. Improved TCP-AP with rt-Winf - TCP-AP with CLI

In a wireless network, nodes along a multi-hop path (NP) contend among themselves for access to the medium, i.e, they contend for available bandwidth.

Considering that TCP-AP only implements adaptive pacing at the sender side, available bandwidth and capacity estimation must take into consideration nodes along the path between the source and the sink, that is, the bandwidth contending successors and predecessors on the route path. Therefore to eliminate this inaccuracy, we changed TCP-AP with *rt-Winf* to use a coefficient (R is the unused bandwidth) that represents the proportion of bandwidth contention among other nodes on the path, thus, maximizing the throughput while guaranteeing fairness. This results in the proposed TCP-AP with *CLI*. If we consider NP as all nodes along the path and if $NP-1$ is equal or less than 4, then TCP-AP with *rt-Winf* is kept unchanged; if $NP-1$ is higher than 4 then the FHD equation, now called the hop delay (HD) is updated to:

$$HD = FHD \times R \quad (5)$$

where

$$R = 1 + \frac{1}{NP} \quad (6)$$

then,

Algorithm 1: *TCP-AP with CLI* Source Node Operations.

foreach *ACK packet* **do**
    Node estimates node path (NP) from MAC ACK
    Node computes NP-1
    if $NP - 1 <= 4$ **then**
        $HD = FHD$;
    else
        $R = \frac{NP+1}{NP}$;
        $HD = R \times FHD$

$$HD = 4 \times (\frac{NP+1}{NP}) \times (t_q + \frac{S_{data}}{AB_{Winf}}) \tag{7}$$

Algorithm 1 shows the pseudo-code of an TCP-AP with *CLI* source node.

As R represents the unused bandwidth due to node contention and queue management along the path, it introduces the fairness factor allowing an improved fair share of the available bandwidth among all contending nodes, not only the ones within the 4-hop propagation delay, improving TCP with *rt-Winf* behavior and making it behave more accurately.

IV. Simulation Results

This section shows simulation results of our proposed congestion control mechanism. The results are obtained using the the ns-2 simulator [14]. The underlying *rt-Winf* mechanism is configured with enabled RTS/CTS/ACK handshake packets. The proposed mechanisms, TCP-AP with *rt-Winf* and TCP-AP with *CLI*, are evaluated against the base TCP-AP protocol and against WCP and XCP-Winf. The network scenario used is an ad hoc network with nodes varying from 8 to 256 nodes (8, 16, 32, 64, 128, 256). Flows also vary according to the number of nodes, with 8 nodes we have 4 flows, with 16 nodes we have 8 flows, and so on. The routing protocol used was the Destination-Sequence Distance-Vector (DSDV) [17]. All simulations last 300 seconds. The simulations were repeated 30 times with different ns-2 seed values. The mean and 95% confidence intervals are presented in the results. The configured default transmission range was 250 meters, the default interference range is 500 meters, and the channel data rate is 11 Mbps. The performance metrics used are: throughput, delay and number of received packets. Each flow presents a FTP application, simulating large file download. The mobility is emulated through the ns-2 *setdest* tool to provide a random node movement pattern. We configure *setdest* 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 [18] scripts adapted to our own needs. To understand how the new proposals behave under different conditions, it was also defined a wireless mesh topology scenario. This scenario is defined with a grid of 16 mesh nodes and a variable number of mobile nodes. The number of mobile nodes changes from 3 to 7. For the data transmissions, it is used a File Transfer Protocol (FTP) application with packets of 1500 bytes. First we analyze the results of the TCP-AP with only the *rt-Winf* information and then the TCP-AP with *CLI*.

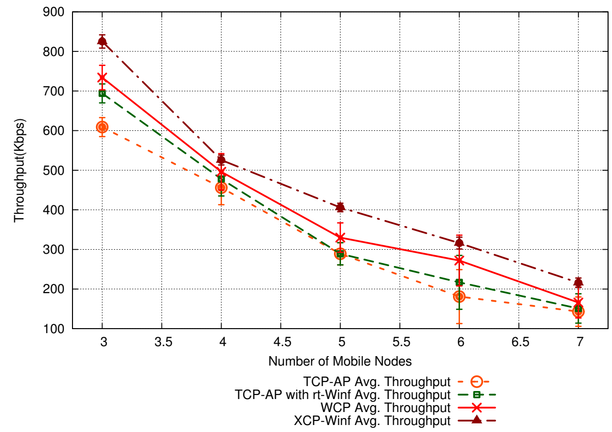

Figure 1. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with rt-Winf Throughput.*

A. *TCP-AP with rt-Winf Results*

Figure 1, Figure 2 and Figure 3 show the performance metrics for the mesh topology scenario. From the observation of the results, it is possible to conclude that TCP-AP with *rt-Winf* information integrated clearly improves TCP-AP performance behavior, but it is below the performance of XCP-Winf. TCP-AP with *rt-Winf* is only taking into consideration *rt-Winf* information for the last four hop nodes; TCP-AP, as oppose to XCP-Winf, uses the standard behavior of TCP for the other hops of the network, considering that all links have the same bandwidth. Another important drawback of TCP-AP with *rt-Winf* is the fact that it does not have a fairness module, resulting in a more conservative and less fair operation. The fairness module is a native mechanism used by XCP-Winf. As TCP-AP with *rt-Winf* uses, in most of its functioning, the standard AIMD process of TCP and is not entirely using the available information, between the source and the sink, on the network, its results are not as good as XCP-Winf. XCP-Winf also relies on total node path interaction, using a cooperative approach to obtain the best available bandwidth and link capacity usage. TCP-AP with *rt-Winf* as the number of nodes, or flows, increases uses conservative mechanisms reducing its performance, especially concerning received packets. The results also show that WCP obtains good results. WCP uses explicit congestion information between nodes that trigger rate changes, making it behave with good efficiency and fairly. As XCP-Winf uses the *rt-Winf* mechanism as its base estimation tool,it has a precise feedback communication mechanism between all the nodes along the path using total network cooperation, it is able to better use the channel with less losses, resulting in a more efficient and accurate behavior,

Figure 4, Figure 5 and Figure 6 show *TCP-AP with rt-Winf* results for the ad hoc topology scenario. It is possible to conclude that *rt-Winf* clearly improves TCP-AP performance, compared to TCP-AP and WCP. However, TCP-AP with *rt-Winf* still reflects some of TCP-AP flaws. With the increase of

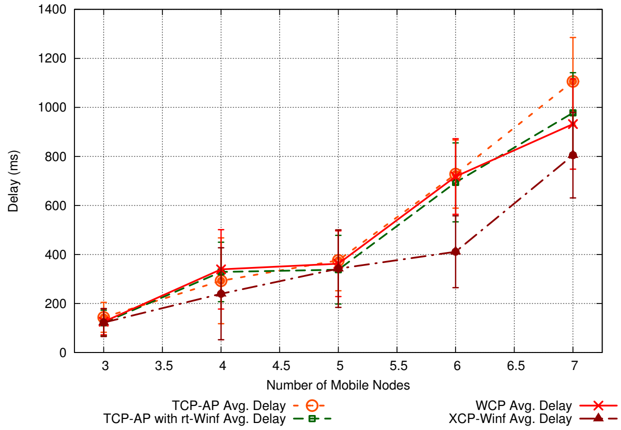

Figure 2. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with rt-Winf Delay.*

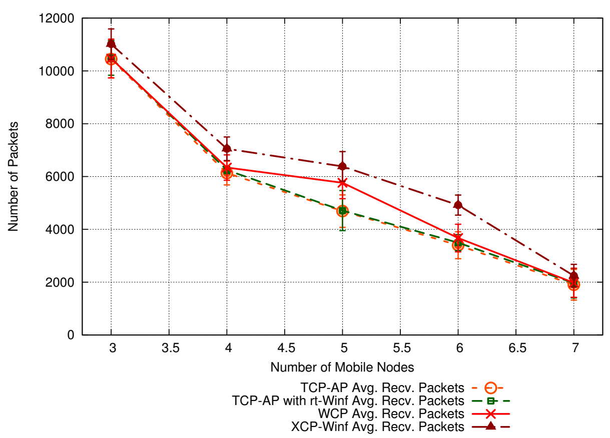

Figure 3. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with rt-Winf Received Packets.*

number of flows, TCP-AP with *rt-winf* becomes less efficient, as it is only relying on the he 4-hop propagation delay and the AIMD process, not considering the entire network topology for its rate changes. This is shown by being able to obtain good throughput results, compared to XCP-Winf, when the network is not heavily loaded. When increasing the number of nodes, number of flows and the mobility density TCP-AP with *rt-Winf* becomes more inefficient, reducing significantly its throughput and number of received packets when compared to the other approaches. TCP-AP with *rt-Winf* is also more fair than TCP-AP to mobility changes, but still shows an unstable behavior. WCP has overall good results, although being an hybrid approach, it uses a more effective congestion and control interval, as all nodes within the congestion neighborhood mark packets with congestion indicators, triggering rate reductions more efficiently at the source.

B. TCP-AP with *CLI Results*

This section presents the simulation results of TCP-AP with *CLI* in the mesh scenario. Figure 7, Figure 8 and Figure 9 show the performance metrics. Figure 7 shows how throughput is improved in TCP-AP with *CLI*. The TCP-AP with *CLI*

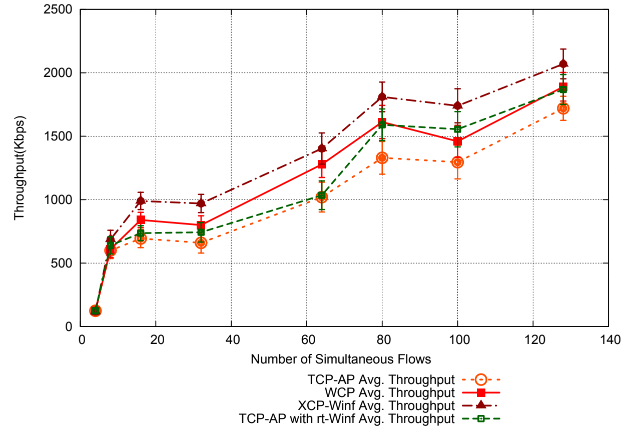

Figure 4. Ad-Hoc Scenario, *TCP-AP* with rt-Winf Throughput.

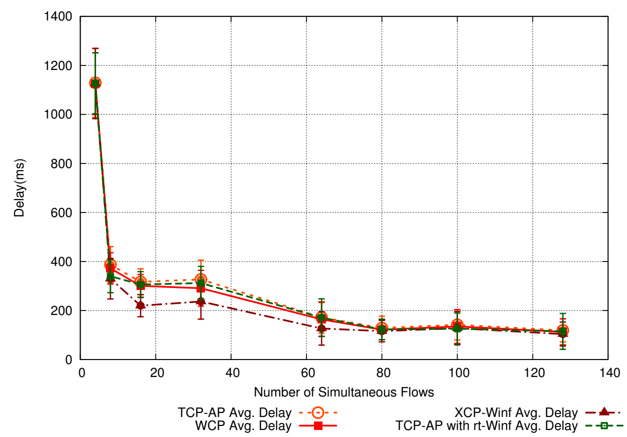

Figure 5. Ad-Hoc Scenario, *TCP-AP* with rt-Winf Delay.

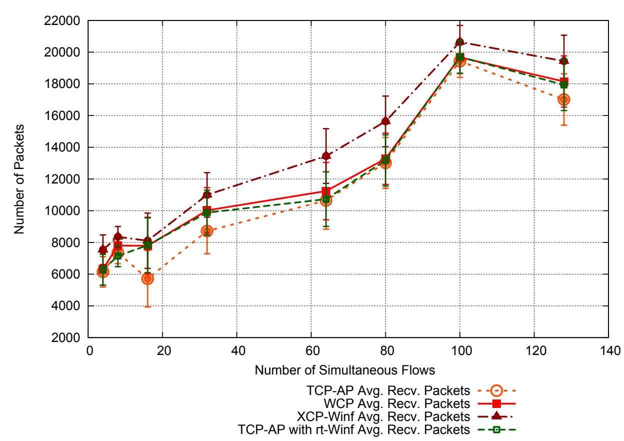

Figure 6. Ad-Hoc Scenario, *TCP-AP* with rt-Winf Received Packets.

throughput values are $\sim$ 20% to $\sim$ 40% better than the ones with the standard TCP-AP, and $\sim$ 14% to $\sim$ 25% better than TCP-AP with *rt-Winf*. In terms of received packets, as observed in Figure 9, it is possible to see that TCP-AP with *CLI* is able to use more fairly the medium, as it is proportionally accounting the unused shared bandwidth, thus, being able to use more efficiently the medium resulting in

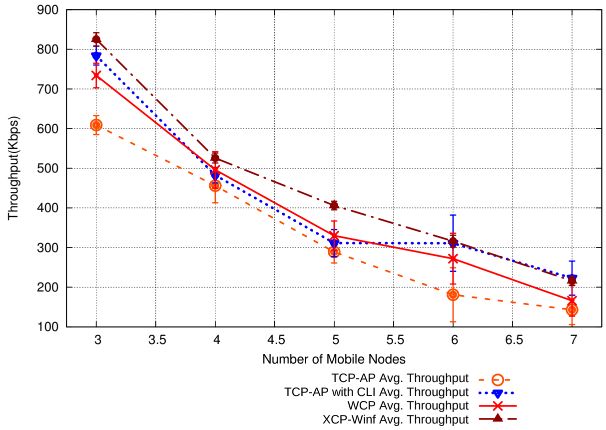

Figure 7. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with CLI* Throughput.

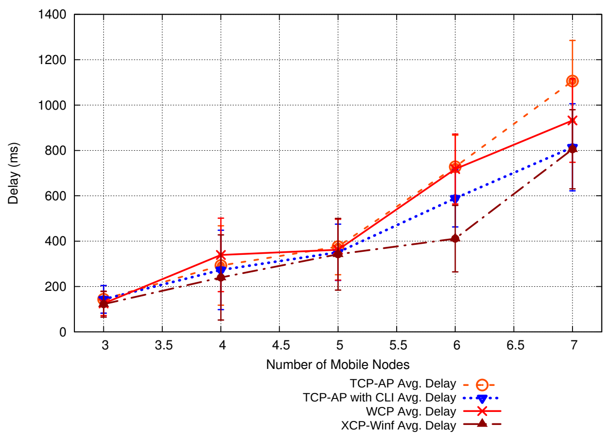

Figure 8. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with CLI* Delay.

more received packets. The network, with these improvements, can transmit with a higher rate and less losses. As more packets are transmitted, more throughput is obtained and the medium is better used. This allows to have a more stable and fair behavior. It is, however, important to say that TCP-AP has a very conservative behavior, as it allows a good throughput with less received packets. This behavior is clearly improved with TCP-AP with *CLI*. The delay values (Figure 8), are also reduced reinforcing the fact that this new proposal is much more efficient and fair, with better medium usage, than the base protocol. The better results are obtained by XCP-Winf, but it is clear that the use of MAC layer information and the node path contention count is making TCP-AP with *CLI* to react more efficiently to the network dynamics.

Figure 10, Figure 11 and Figure 12 show TCP-AP with *CLI* results in the ad hoc networks scenarios, as defined before. From the observation of the results it is possible to infer that TCP-AP with *rt-Winf* integrated, the node path contention and collision probability clearly improves base TCP-AP performance behavior. It is possible to conclude that, with more nodes and flows in the network, TCP-AP with *CLI* is more efficient than the standard TCP-AP proposal. XCP-

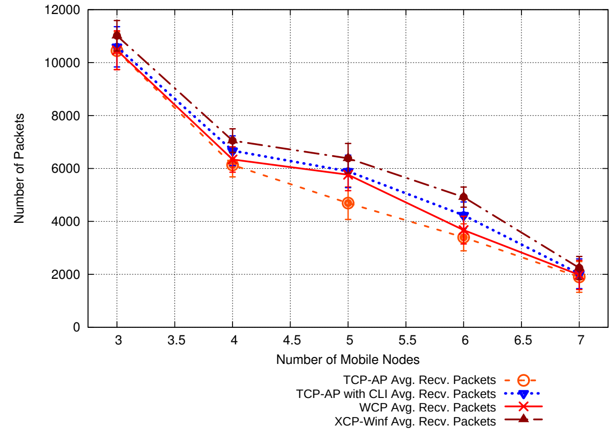

Figure 9. 16 Mesh Nodes - Variable Number of Mobile Nodes, *TCP-AP with CLI* Received Packets.

Winf uses a explicit congestion control notification mechanism for an accurate rate change and relies in link capacity MAC layer information, is able to operate more efficiently than TCP-AP with *CLI*, specially concerning the number of received packets. XCP-Winf is more fair and obtains better medium usage. TCP-AP with *CLI* still uses for most of its operations the standard TCP congestion window process, not using, as XCP-Winf, a explicit congestion notification mechanism that allows to react more quickly to the network changes. XCP-Winf, as is based in the base XCP protocol, also has a fairness module that allows it to be more accurate and efficient.

TCP-AP with *CLI*, as opposed to TCP-AP, is considering a fair share of the unused bandwidth, that results from the use of the node path contention count, making it behave more efficiently and allowing it to increase the flow rate, and consequently increase the number of received packets and reducing the overall delay. We can conclude that the available bandwidth and capacity evaluation of *rt-Winf*, estimated at the MAC layer, the collision probability and the node contention count factors are relevant and surely make TCP-AP with *CLI* behave more consistently and with better channel utilization, which also leads to less channel losses. Comparing both ad hoc and mesh results, it is evident that TCP-AP with *CLI* results are better on the ad hoc environments; this is due to the fact that TCP-AP was developed having in mind ad hoc networks and, also, due to the fact that its underlying hybrid scheme is better suited for ad hoc networks. Moreover, its underlying hybrid scheme is better suited for ad hoc networks with high density and mobility.

From the presented results it is also possible to observe that WCP has better overall results than TCP-AP. WCP has a rate control mechanism that reacts explicitly to congestion, and a cooperative communication process between neighbor nodes that make WCP to react more efficiently to the network conditions, allowing to have a better medium usage.

We can also conclude that TCP-AP is not using efficiently the medium, resulting in poor performance results. TCP-AP is not evaluating correctly the available bandwidth obtaining poor throughput and behaving very conservatively due to the AIMD process, resulting in low received packets results and

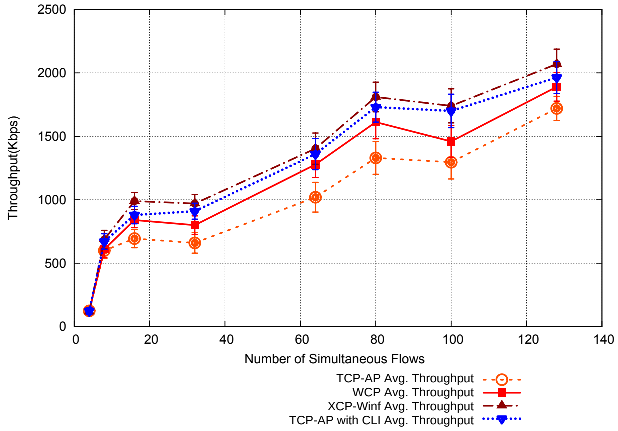

Figure 10. Ad-Hoc Scenario, *TCP-AP with CLI* Throughput.

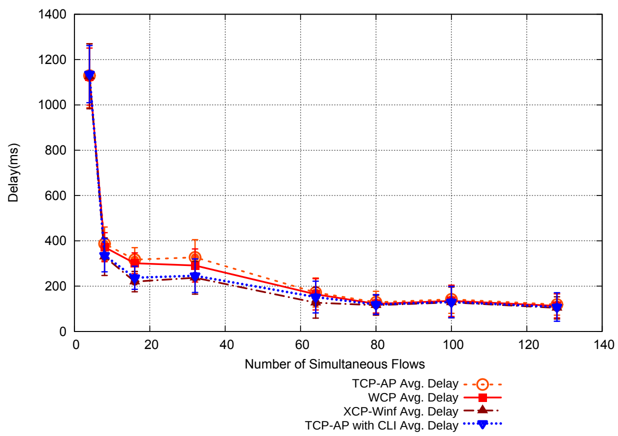

Figure 11. Ad-Hoc Scenario, *TCP-AP with CLI* Delay.

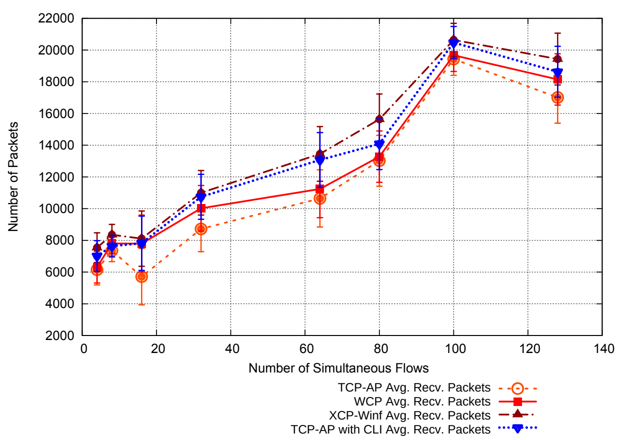

Figure 12. Ad-Hoc Scenario, *TCP-AP with CLI* Received Packets.

higher delay. TCP-AP is not considering a fair share of the bandwidth to all flows, not using correctly the medium and having a significant degradation of performance.

Although TCP-AP scheme is a hybrid scheme of sender rate control and congestion control, TCP-AP is based on two assumptions, that the rate control mechanism is efficient and the contention and spatial reuse is accurate, whether they are

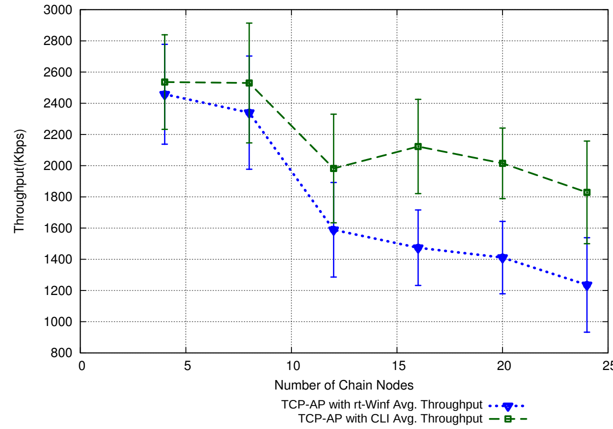

Figure 13. Chain Scenario, Throughput.

effective in some network topologies remains unknown. This assumption is clearly not effective in high mobility wireless scenarios.

For a better understanding of how the factor R (Equation 6) is influencing TCP-AP with *CLI* behavior, a central network chain scenario was defined. It must be noted that the standard TCP-AP 4-hop propagation delay assumes that "every fourth node can transmit in a multi-hop chain topology". On this scenario, it was used the proposed version of TCP-AP with *CLI* and the TCP-AP with *rt-Winf* version. The chain scenario consists of a network divided in three parts. The central part contains chain nodes. A left and a right lateral sides contain four sending nodes and four receiving nodes. The application used simulates a FTP transfer. The results are shown on Figure 13, Figure 14 and Figure 15. The presented results clearly show that with the increase of the chain nodes TCP-AP with *rt-Winf* becomes less efficient an less accurate, as it is not considering the unused share of bandwidth being more unfair. TCP-AP with *CLI* is more accurate as the available bandwidth and capacity estimation are considering the nodes along the path between the sources and the sinks nodes, that is, the contending successors and predecessors on the route path. It is, then, proved that the factor R, which represents the proportion of bandwidth contention among other nodes on the path, is maximizing the throughput while guaranteeing fairness.

C. Utility Results

As TCP is the most used and deployed congestion control protocol on the Internet, it is important, as described on [19], to analyze how TCP-AP with *CLI* flows interact and compete with TCP. For analyzing how friendly TCP-AP with *CLI* is, we use the average data rate over time for each flow, thus, allowing to observe how bandwidth is being managed between TCP and the TCP-AP with *CLI* proposal. This is called the utility of a congestion control mechanism against TCP. The evaluation scenarios consist of a 1000mx1000m area, divided on three distinct parts. In the left side area, with 250mx250m, we have two mobile source nodes. One source node configured to use only the standard TCP, and the other source node with the TCP-AP with *CLI* congestion

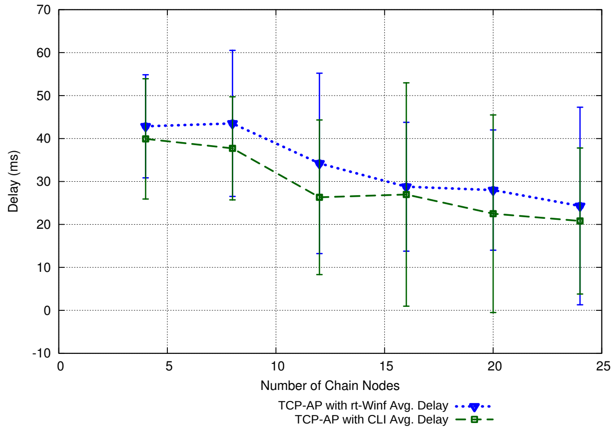

Figure 14. Chain Scenario, Delay.

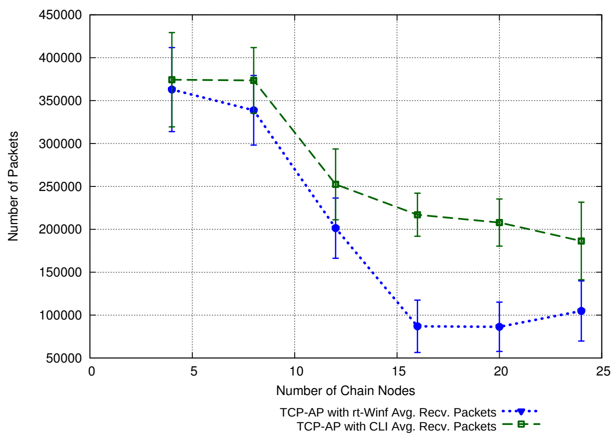

Figure 15. Chain Scenario, Received Packets.

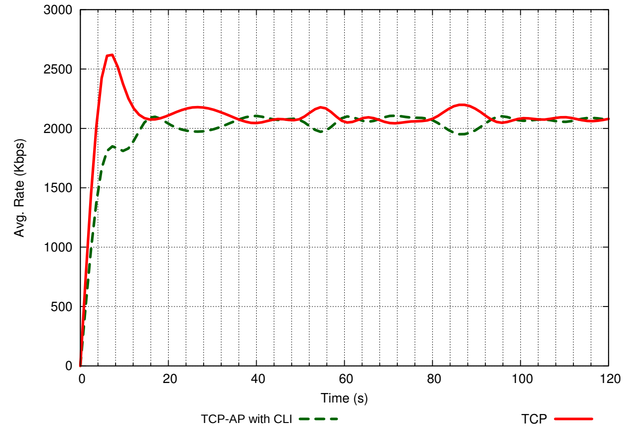

Figure 16. Utility Results, 2 x 8 Flows.

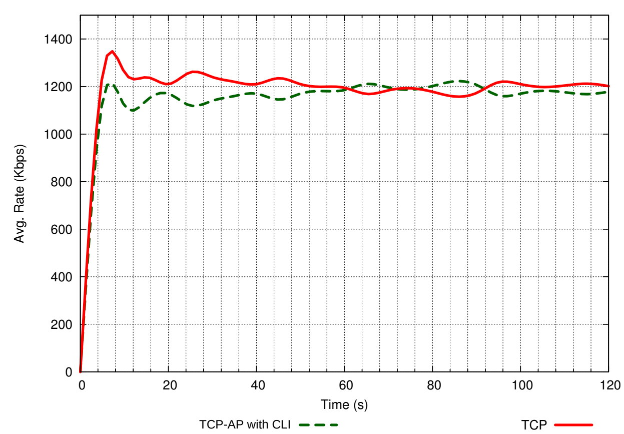

Figure 17. Utility Results, 2 x 16 Flows.

control mechanism configured. The right side of the area has the same characteristics of the left are but instead of source we have sink nodes. Finally, the middle area, with 500x500m, has two mobile nodes configured with the TCP-AP with *CLI* mechanism as their main congestion control mechanism, the average data rate is measure on this two nodes, as they will have TCP and TCP-AP with *CLI* like flows competing. We have defined two evaluation scenarios. In one scenario we have each source generating eight FTP flows, with packets of 1500 bytes. In the other scenario we have each source generating sixteen FTP flows. The simulations last 120 seconds. The obtained results are shown on Figure 16 and Figure 17.

From the utility results, it is possible to observe that, on both situations, the TCP flow grows faster and gains more on the beginning. However, as TCP-AP with *CLI* is an hybrid approach, keeping unchanged the AIMD process of TCP, and is updated with an evaluation and measurement process, it quickly adjusts to TCP behavior, thus, allowing a fair share of network resources.

V. Conclusions and Future Work

This paper proposed a new approach to congestion control, based on TCP-AP and a wireless inference mechanism, called *rt-Winf*. *rt-Winf* measures the wireless capacity and the available bandwidth of wireless links, and feeds this information to TCP-AP, through a cross layer communication process. Two different improvements were also considered on the new approach, the introduction of collision probability on the available bandwidth approach and the node path contention count on the 4-hop propagation delay approach.

The performance evaluation study of the proposed congestion control mechanism shows that the integration of *rt-Winf* and the mentioned improvements, allow TCP-AP behavior to be more efficient and effective, resulting in better overall network performance. Using *rt-Winf*, that works in the MAC layer, it is possible to perform link capacity and available bandwidth calculations without interfering in the network dynamics, allowing to significantly improve TCP-AP performance. The results also show, however, that as TCP-AP with *rt-Winf* still relies in the main principles of TCP-AP, specially in the 4-hop propagation delay for its estimations. Its results are not accurate resulting in bad medium usage and in a conservativeness behavior. Considering for the available bandwidth and capacity estimation the nodes along the path that contend for available bandwidth makes TCP-AP to be more fair, to use more efficiently the medium resulting in good network performance results. This final congestion control mechanism based in TCP-AP is defined as TCP-AP with *CLI*.

As future work, we plan to work on the wider evaluation of the congestion control approach, using, for example, new comparison baselines and protocols. An effort will also be made in creating a future test bed for understanding how the proposed mechanism is affected by different conditions and parameters, in a real environment.

References

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

[2] 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.

[3] S. M. ElRakabawy, A. Klemm, and C. Lindemann, "TCP with pacing for multihop wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '05. ACM, 2005, pp. 288–299.

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

[5] L. Barreto, "XCP-winf and RCP-winf: Improving explicit wireless congestion control," *Journal of Computer Networks and Communications*, vol. 2015, no. Article ID 925207, pp. 1–18, 2015.

[6] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan, "Understanding congestion control in multi-hop wireless mesh networks," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08. ACM, 2008, pp. 291–302.

[7] K. young Jang, K. Psounis, and R. Govindan, "Simple yet efficient, transparent airtime allocation for tcp in wireless mesh networks," in *6th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2010. [Online]. Available: http://www-bcf.usc.edu/∼kpsounis/Papers/cna.pdf

[8] M. Li, D. Agrawal, D. Ganesan, and A. Venkataramani, "Block-switched networks: A new paradigm for wireless transport," in *6th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI 2009)*, April 2009.

[9] A. Aziz, D. Starobinski, P. Thiran, and A. E. Fawal, "Ez-flow: Removing turbulence in ieee 802.11 wireless mesh networks without message passing," in *5th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2009.

[10] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing tcp fairness in ad hoc wireless networks using neighborhood red," in *Proc. ACM MobiCom*. ACM Press, 2003, pp. 16–28.

[11] F. Abrantes and M. Ricardo, "A simulation study of XCP-b performance in wireless multi-hop networks," in *Proceedings of the 3rd ACM workshop on QoS and security for wireless and mobile networks*, ser. Q2SWinet '07. ACM, 2007.

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

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

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

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

[16] *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

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

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

[19] J. F. L. Xu, "TCP-friendly CBR-like Rate Control," *Network Protocols, 2008. ICNP 2008*, pp. 177 – 186, October 2008.