

rt-Winf: Real Time Wireless Inference Mechanism

Bruno Rés¹, Luís Barreto², Susana Sargento¹

¹ Instituto de Telecomunicações, Universidade de Aveiro, Portugal

{bruno.res,susana}@ua.pt

² Instituto Politécnico de Viana do Castelo, Portugal

lbarreto@esce.ipv.pt

Abstract—This paper presents a new method to estimate the available bandwidth and the path capacity over a wireless network path, denoted as rt-Winf. 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. rt-Winf repeatedly samples the available bandwidth of the network path, requiring little computation in each iteration, and being lightweight with respect to memory requirements. rt-Winf has been tested both in the CMU Wireless Emulator and the ns-2 simulator, using ad-hoc, infrastructure and mesh networks scenarios. The results obtained through the comparison of rt-Winf with current state of the art approaches show that rt-Winf obtains the available bandwidth and capacity estimation with the highest accuracy and without introducing overhead traffic in the network.

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

I. INTRODUCTION

Nowadays, there is an exponential use of wireless computing platforms for Internet access and file sharing among different users. Thus, it is important to know, in real time, some major network properties. One of such properties is the available bandwidth, which can be used for congestion control, routing and scheduling decisions, leading to quality improvement, support of real time services, and more efficient wireless link usage.

Available bandwidth and path capacity estimation have been widely studied, but most of the mechanisms work in scenarios with wired and/or last hop wireless networks. Tools like [1][2][3][4] are some of those examples. These tools work sending out a series of various probe packets with different sizes and, for each probe, they measure the time an error packet is received. The bandwidth of each link and its latency are obtained through statistical analysis of those measurements. Other tools like AbGet [5], PathChirp [6], PathLoad [7] rely on self induced congestion. They use the principle of injecting probe packets in the network, with some inter-packet separation, and measuring the effects of these packets. When a probe packet arrives at the receiver, it is timestamped. The inter-packet separation is increased if the packets have suffered congestion on the path. These mechanisms are also denoted active measurement methods. Ad Hoc probe [8] is a wireless active measurement technique that uses packet pairs to measure the end to end path rate based on one way measurements.

IdleGap [9] is different from the previous approaches as it is a new mechanism to infer the available bandwidth in a passive way, without the need to use probe packets. Accordingly to [9], cross layer feedback is performed by a device accessing its own protocol stack layer that contains information from the transmitted packets. IdleGap adds a new module between the Media Access Control (MAC) and network layer to the nodes system, altering its protocol stack. Moreover, its results are not accurate. In this paper we present a new available bandwidth mechanism, based on IdleGap, that does not require a new module in the system, thus the protocol stack is not altered. As opposite to IdleGap, our mechanism is accurate, as it determines the real capacity instead of using the IEEE 802.11 header data rate. Our mechanism is denoted as rt-Winf - Real Time Wireless Inference Mechanism. rt-Winf measures the available bandwidth and link capacity through the information included in the RTS/CTS packets, measuring the effective transmission time of the existent traffic, through the estimation of each node's channel allocation. In case RTS/CTS are not enabled, rt-Winf is able to perform the same operation with a small amount of probing packets. The results of rt-Winf show that the capacity and available bandwidth are correctly estimated and that the interference and overhead included in the network are very small (smaller than 1%).

The rest of this paper is organized as follows. Next section, Section II, briefly presents the related work. Then, section III describes the rt-Winf algorithm. Section IV describes and discusses the results obtained through emulation and simulation. Finally, section V presents the conclusions.

II. RELATED WORK

Link capacity estimation has been widely studied in wired networks. PathChirp [6], IPerf [10], Pathchar [2], CapProbe [3] are some examples. AbGet [5], Pathload [7] and IGI/PTR [11], are some examples of available bandwidth estimation methods. There are also developments with respect to wireless networks, such as AdHoc Probe [8], WBest [12] and IdleGap [9]. AdHoc Probe provides only the path capacity of the wireless channel. WBest calculates both capacity and available bandwidth.

WBest contains an algorithm that is divided in two phases. In the first one, it uses packet pair techniques in order to determine the capacity. In the second phase, it uses packet train techniques in order to determine the available bandwidth. In fact, in this phase, packets are sent at the rate obtained in the first phase. It means that in this period of time the WBest

tool is being very intrusive, causing undesired problems in the network.

IdleGap is a recent mechanism for obtaining available bandwidth in wireless networks. It is not realistic in the determination of link capacity and introduces a new sublayer in the model stack. However, it uses a very accurate approach to characterize the busy time and the total elapsed time, obtaining a very accurate *Idle Rate*. Thus, our algorithm was based in this mechanism, which will be better described below.

A. IdleGap

This method is focused in the CSMA-CA scheme of wireless networks. It takes Network Allocation Vector (NAV) [13] into consideration. NAV represents the time that other nodes will occupy the medium. It is used by the idle nodes which are waiting to transmit.

The Available Bandwidth (AB) is calculated using the occupation time:

$$AB = IdleRate * DataRate \quad (1)$$

The *DataRate* value is obtained from the IEEE802.11 [14] packet header. 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*, where the *Sender* is transmitting data; the *Receiver* is receiving data and, finally, the *Onlooker* is not participating in the transmission. The *Idle Module* is used to determine the *Idle Rate*,

$$IdleRate = 1 - \frac{Busy\ Time}{Total\ Elapsed\ Time} \quad (2)$$

where *Busy Time* is equal to $\sum NAV$. The authors consider 10 seconds intervals for obtaining the total elapsed time.

The introduction of the *Idle Module* has an important disadvantage, that is the modification, by the introduction of a new sublayer, of the OSI Model [15]. The *Idle Module* is responsible for the determination of the *Idle Rate*, differentiating the existing traffic in three categories, according to its destination: to the node, from the node and between other nodes. All existing wireless network equipments cannot use this method, as they need to be modified in order to have this sublayer. This is an important issue, as it will introduce incompatibility problems.

IdleGap uses the pre-defined IEEE802.11 header *DataRate* value, which is not practical and real, thus leading to not very accurate and over-dimensioned estimation values. The *DataRate* value used can be considered a maximum theoretical value. Even in ideal conditions, this maximum value will never be reached, due to overhead caused by packet headers, time and Quality of Service (QoS) constraints. This difference between the real and the used *DataRate* is introducing a systematic calculation error in IdleGap, leading to its lack of accuracy. We intend to improve the results obtaining accurately the capacity value without the need to modify the OSI model stack.

For this purpose, we will make use of Request To Send (RTS) and Clear To Send (CTS) packets present in the

Carrier Sense Multiple Access - Collision Avoidance (CSMA-CA) [16] to correctly retrieve the Network Allocation Vector (NAV). We will then use this value to determine the capacity and available bandwidth. The usage of RTS/CTS scheme is optional, but it is nowadays widely supported by all wireless equipments. Its usage, in a traditional wireless network, represents a negligible cost in terms of overhead [17]. This may be seen as a very important advantage of using RTS/CTS packets.

III. RT-WINF

The rt-Winf mechanism was developed inspired by IdleGap [9], but with the purpose to mitigate the problems previously mentioned, being compatible with all systems and determining both the link capacity and available bandwidth without overloading the network. These characteristics overstep AdHoc Probe and WBest inherent limitations and problems. rt-Winf does not introduce any change to the OSI Model, as opposed to IdleGap, being able to obtain all the necessary times to calculate the path capacity and available bandwidth. Another important aspect of rt-Winf, relatively to IdleGap [9], is the effective calculation of the capacity, instead of using the *DataRate* value of the IEEE802.11 header [14].

A. RTS/CTS Packets

rt-Winf with RTS/CTS control packets enabled relies on this handshake to correctly retrieve the NAV values.

In order to evaluate the accuracy of the duration field on the IEEE802.11 header we performed a large number of captures (~ 200). We concluded that the duration value on data packets is not reliable, because different sized packets have always the same duration. The RTS/CTS packets have accurate duration values, which can be used in the calculations.

With the obtained captures, it was possible to realize how each state managed the received packets. In the case of the *Sender* state, the node was able to capture the CTS, DATA and ACK packets. A node in the *Receiver* state was able to capture the RTS and the DATA packets, while a node in the *Onlooker* state was able to capture the complete set of packets: RTS, CTS, DATA and ACK. This different knowledge implied the conception of different algorithms for each state. Then, we propose that each node state uses a different method to determine the *Idle Rate*. In the case of the *Sender*, it is considered the NAV of the CTS packets on the available bandwidth calculation. For the capacity calculation, it is considered the time that the channel is busy, that is, the difference between ACK time, CTS time and the duration of the occurred Short Inter-Frame Spacing - SIFS (where ACK time is the actual clock time when the ACK packet is Received, and CTS time is the clock time when CTS packet is received). The *Receiver* uses the NAV of the RTS packets to obtain the *Idle Rate* and the difference between the DATA time, RTS time and 3 times SIFS to obtain the capacity (where DATA and RTS times are, respectively, the clock time when DATA packet is received and RTS packet is received). The *Onlooker* uses the NAV value according to the existence, or not, of the RTS packet to obtain both the available bandwidth and capacity. If a node in the *Onlooker* state captures a CTS packet of a communication without capturing the RTS

packet, this implies that the communication is suffering from the *hidden nodes* problem. Thus, the algorithm will only use the NAV from the CTS packet to retrieve the correct values. The total elapsed time represents the difference between the last captured ACK time and the initial time. The packet size considered is the DATA packet size. Figure 1 shows the different approaches for each state while Figure 2 represents the state diagram of the rt-Win tool. It is possible to observe each state's transitions. When a node is not transmitting or receiving packets it is on the *Onlooker* state. In this state, the node calculates the onlooking capacity. Thus, it can use this information, when changing to the *Sender* or *Receiver* state. The onlooking capacity is obtained as described in Figure 1. When a CTS packet is captured by the *Sender*, it starts to evaluate the available bandwidth and capacity, while the *Receiver* starts this process when a RTS packet is received. The *Receiver* sends the calculated available bandwidth and capacity in an ACK packet to the *Sender*. When the *Sender* receives the ACK packet with that information, from the *Receiver*, compares it with the available bandwidth and capacity that it has previously calculated. If the information received through the ACK packet is lower than the obtained, the *Sender* will use the available bandwidth and capacity received in the ACK packet. Otherwise, the *Sender* will transmit using the available bandwidth and capacity calculated before. This cooperation is a great improvement when compared to IdleGap.

B. Probe Packets

If RTS/CTS packets are not present, rt-Win can use probe packets in order to retrieve the transfer time values. Probe packets can be sent between nodes. These must be UDP generated packets with altered Frame Control IEEE 802.11 header: Type Data and Subtype Reserved. We used packets with Frame Control Type set to 10 (data) and Subtype to 1001 (Reserved). This 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 [14]. 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 3 and Equation 4. These packets are only sent before a node wants to start a transmission and in absence of traffic. This allows the system to initially determine the available bandwidth and capacity. Then, the existing traffic and the MAC layer ACK will be used to trigger the calculations. As NAV values are not correctly defined in DATA packets, rt-Win uses clock time information to determine the busy time. So, NAV values are not considered in this specific implementation with probe packets. To be fully operational, both Sender and Receiver must be running the rt-Win mechanism.

$$C = \frac{PacketSize}{TransferTime} \quad (3)$$

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

$$AB = 1 - \left(\frac{\sum TransferTime}{TotalElapsedTime} \right) * C \quad (4)$$

State	Available Bandwidth	Capacity
<i>On-looker</i>	Captured RTS Packet? YES: $AB = C \left(1 - \frac{\sum NAV_{RTS}}{Total\ elapsed\ time} \right)$ NO: $AB = C \left(1 - \frac{\sum NAV_{CTS}}{Total\ elapsed\ time} \right)$	$C = \frac{Packet\ Size}{ACK_{Time} - CTS_{Time} - 2SIFS}$
<i>Sender</i>	$AB = C_{Sender} \left(1 - \frac{\sum NAV_{CTS}}{Total\ elapsed\ time} \right)$	$C_{Sender} = \frac{Packet\ Size}{ACK_{Time} - CTS_{Time} - 2SIFS}$
<i>Receiver</i>	$AB = C_{Receiver} \left(1 - \frac{\sum NAV_{RTS}}{Total\ elapsed\ time} \right)$	$C_{Receiver} = \frac{Packet\ Size}{DATA_{Time} - RTS_{Time} - 3SIFS}$

Figure 1. rt-Win Algorithm.

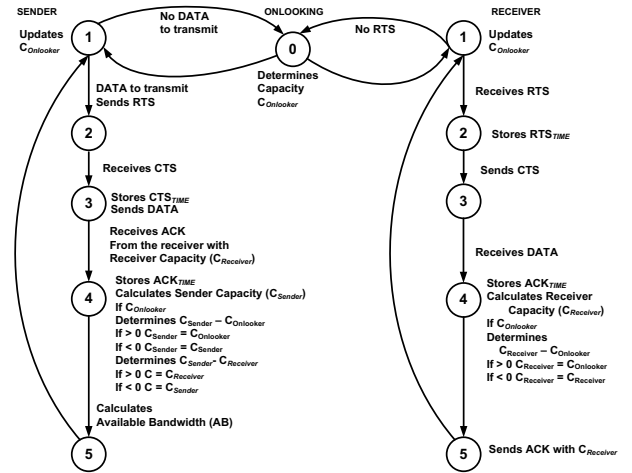


Figure 2. rt-Win Sender, Receiver and Onlooking State Diagrams.

In a normal VoIP call using G.711 codec [18], 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. SIMULATION RESULTS

This section shows simulation results of our proposed mechanism to evaluate the available bandwidth and path capacity on a set of wireless network scenarios. The results are obtained using the CMU Wireless Emulator [19] and the ns-2 simulator [20]. Although the emulator provides more realistic results than a traditional network simulator, we also present ns-2 simulations for comparison purposes. In base rt-Win, the system is configured with enabled RTS/CTS/ACK handshake packets. In rt-Win probe, RTS/CTS/ACK handshake is not enabled, and probe packets are implemented, the maximum achievable data rate is set to 11 Mbps. Nodes are placed in such a distance that the path loss effect is considered negligible. The three states defined by rt-Win mechanism and the cooperation between them and between the nodes was developed in C language. Several scenarios were used, varying the number of nodes and the traffic load. Most of the scenarios used an Access Point (AP), for the management of the wireless communication. An ad hoc and a mesh wireless scenario were also considered to evaluate rt-Win.

A. Path Capacity

The results for the path capacity estimation are presented in this section. We compared our results with AdHoc Probe tool and maximum throughput (that represents the maximal

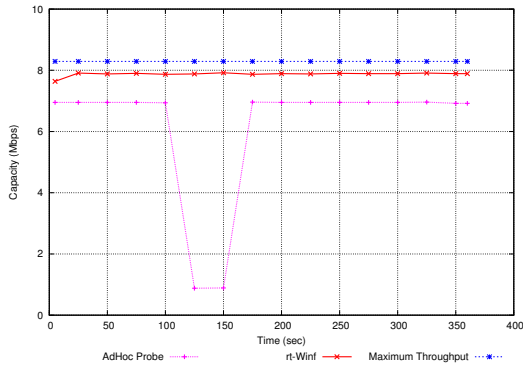


Figure 3. AdHoc Probe and rt-WinF Path Capacity Estimation.

theoretical throughput). AdHoc Probe tool measures efficiently the path capacity in a wireless communication [8] when compared to other tools.

Figure 3 presents the path capacity results obtained with rt-WinF and AdHoc Probe. The testbed used was composed by 2 ad hoc nodes. Each mechanism was tested separately, with the same conditions and duration. An UDP flow with Constant Bit Rate (CBR) of 64 Kbps was injected between the two nodes. Accordingly to [21], it is possible to obtain the maximal theoretical throughput, through

$$TH_{80211b} = \frac{MSDU}{Delay_per_MSDU} \quad (5)$$

where $Delay_per_MSDU$, in seconds is obtained by

$$(T_{DIFS} + T_{SIFS} + T_{BO} + T_{CTS} + T_{ACK} + T_{RTS} + T_{DATA}) * 10^{-6} \quad (6)$$

In equation 6, T represents the duration of each event in μs . Using 1500 bytes as MAC Service Data Unit (MSDU)¹ and measuring the $Delay_per_MSDU$, we obtained the maximal theoretical throughput values.

In ideal conditions, the maximum throughput represents the maximum achievable capacity. As we are using a low CBR flow, the expected capacity should be less than the maximum throughput. This was validated in the simulations, with rt-WinF results being close to the maximum throughput ones. It is possible, then, to observe that rt-WinF uses efficiently the information present in the channel, in order to obtain the resulting capacity. This is because rt-WinF measures more accurately the channel occupation time, as it takes into consideration all traffic flows. Comparing with the AdHoc Probe method, and with a similar probing time, rt-WinF gathers more information to perform the desired calculations, thus being able to be statistically more precise and less sensitive to flow variations. AdHoc Probe only takes into consideration its probing packets, which with the network dynamics can suffer collisions, introducing a negative impact in the capacity calculation.

B. Available Bandwidth

This section presents the obtained results for the channel available bandwidth, comparing with IdleGap [9], IPerf [10],

¹Is received from the logical link control (LLC) sub-layer which lies above the medium access control (MAC) sub-layer, in the protocol stack.

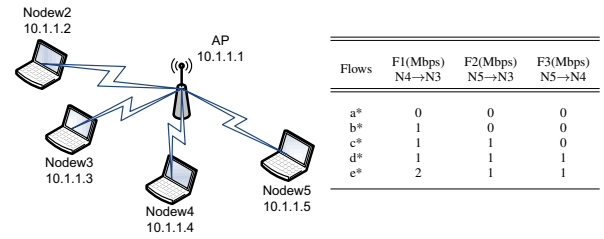


Figure 4. Available Bandwidth Estimation Scenario and Specification.

maximum throughput and maximum achievable data rate (11Mbps) values.

According to [12], the ground truth of the achievable available bandwidth of a dynamic Wireless Local Area Network (WLAN) is approximated the downstream throughput of a single saturated CBR UDP flow with a packet size of the Maximum Transmission Unit (MTU). Then, an UDP flow with a very high constant bit rate (10Mbps) was considered and injected through the channel. The result of the downstream throughput is the curve "IPerf UDP". For the "IPerf TCP" results, we used its default parameters, only changing the sampling time.

The "Maximum Throughput" is obtained from equation 5, which corresponds to the maximal theoretical throughput for each scenario.

Once more, rt-WinF results are closer to the ones obtained by equation 5.

Figure 4 shows the used testbed. Different scenarios contain different cross-traffic flows. Each scenario is labeled and differs on the concurrent traffic flow and number of used nodes. A value of zero represents nonexistence of the node in the considered scenario. We conducted 50 emulations for each scenario, resulting in a confidence interval of 95%.

In Figure 5, we show the variation of available bandwidth as a time function, compared with both the maximum achievable data rate and the maximum theoretical throughput. It is possible to observe the results provided by each tool. The maximum theoretical throughput does not take in consideration the cross traffic influence. Then, it can be considered a valuable information as an upper bound of the expected results. On the other hand, IPerf UDP values are obtained through the overloading of the wireless channel, thus being considered a lower bound of the expected results. As rt-WinF obtains results within those bounds, we can conclude that the determined values are nearer the real ones, and then, rt-WinF is more precise. This is due to the fact that this tool actually measures the channel occupation time of each packet, thus obtaining more precise values for the capacity. Available bandwidth, as described in section III, depends directly on the channel capacity. The IPerf UDP values are close to the ones obtained with rt-WinF, but they are obtained through a very intrusive way. As expected, the IPerf TCP results are the worse, as a result of its congestion control mechanisms, namely, congestion avoidance and slow start [22]. This is represented in Figure 5 by IPerf TCP variations.

IdleGap values are larger than the ones obtained with other tools and are very close to the maximum *DataRate*. To observe this fact, maximum *DataRate* values are also shown in Figure

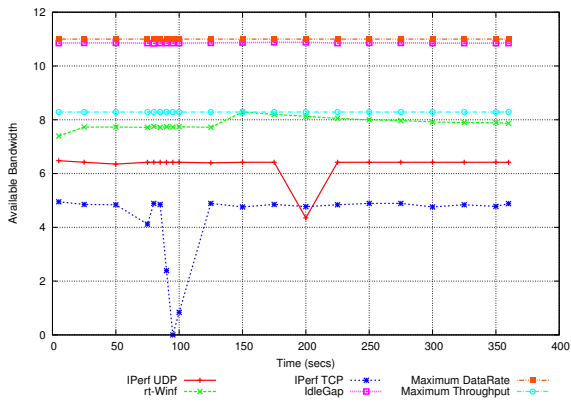


Figure 5. rt-Winf vs IdleGap vs Iperf Available Bandwidth.

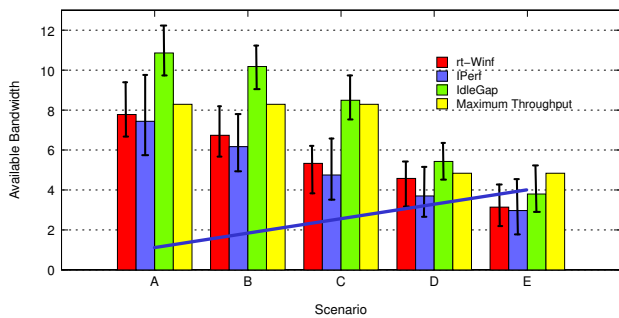


Figure 6. Available Bandwidth by Scenario

5. However, this fact does not mean accuracy. This is due to the fact that this tool, as described in section II-A, does not determine the real available bandwidth. IdleGap obtains this value from the *DataRate* field of each packet. Although this value is the used IEEE 802.11 standard *DataRate*, it is not the real time channel capacity. This value is defined by the standard [14] and varies, only, accordingly with signal propagation issues and network traffic. The standard allows a predefined set of data rate values to determine, in real time, the channel capacity.

Each bar in Figure 6 represents the average emulation results for each scenario (Figure 4). We compare the results provided by IPerf UDP, IdleGap and rt-Winf. The line in the figure represents the variation of the concurrent traffic on the considered scenario. The concurrent traffic varies from 1 Mbps to 4 Mbps.

From the Figure 6, it is possible to observe that IdleGap results are higher than rt-Winf and IPerf. Again, this is related with the fact described before. IdleGap values vary in real time and IdleGap evaluates the occupied time in a similar way as rt-Winf. However, when this occupation rate is obtained and is applied with the IEEE802.11 header *DataRate* value, the resulting available bandwidth is being erroneously magnified. As noted on Figure 6, with the increase of concurrent traffic, the difference of values between IdleGap and the other tools is significantly reduced. This is due to the fact that, for more concurrent traffic, the device drivers automatically adjust the data rate value to a lower one. In scenarios A, B and C it is set to 11Mbps, and in scenarios D and E, it is adjusted to 5.5 Mbps.

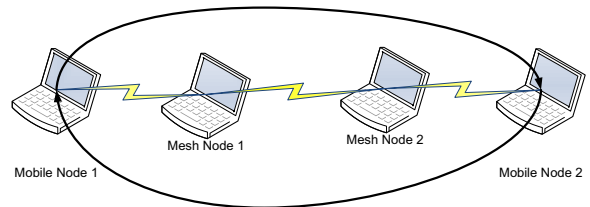


Figure 7. Wireless Mesh Scenario

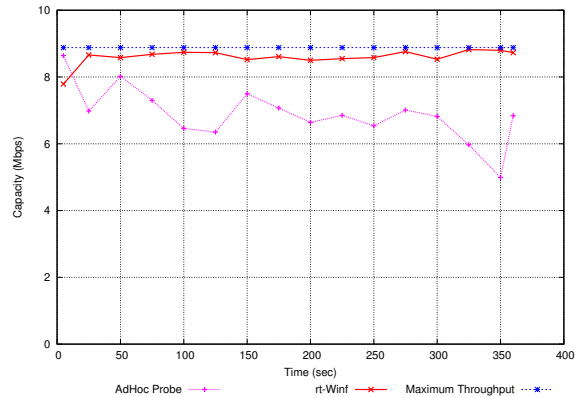


Figure 8. Wireless Mesh Scenario Path Capacity

C. Wireless Mesh Network Results

Path capacity and available bandwidth were also evaluated on a wireless mesh scenario. The testbed used is shown in Figure 7. It uses two mobile nodes, *Mobile Node 1* and *Mobile Node 2*, communicating with each other through two mesh nodes responsible for the routing and link management. The mobile nodes are in such a distance that the traffic is routed by the mesh nodes. In this scenario there is no concurrent traffic and each simulation was conducted independently. The residual traffic on the network is introduced by each mechanism.

The obtained results of path capacity are shown in Figure 8, and the results of available bandwidth are shown in Figure 9. In this figure, we show the results provided by rt-Winf, IPerf UDP and IdleGap. Maximum throughput values are also presented, being considered as an upper bound of the result, as described before. The lower bound is, once more, the IPerf UDP result.

As observed in Figure 8, rt-Winf is less sensitive to variations when compared to AdHoc Probe. This is because rt-Winf is taking into consideration all packets in the network and is measuring the channel occupation time of each packet, while AdHoc Probe is only considering the packets that it generates, thus, being more sensitive to flow variations.

Through the results in Figure 9, it is possible to observe how IdleGap is not effectively measuring the available bandwidth. IdleGap values have a small variation, but are near the *DataRate* value, which is also higher than the maximum achievable throughput, and is not taking into consideration the network conditions. As opposed to IdleGap, rt-Winf provides more real results, as it is possible to observe how the results vary through time. Those results are within an upper bound, the maximum theoretical throughput, and a lower bound, IPerf UDP.

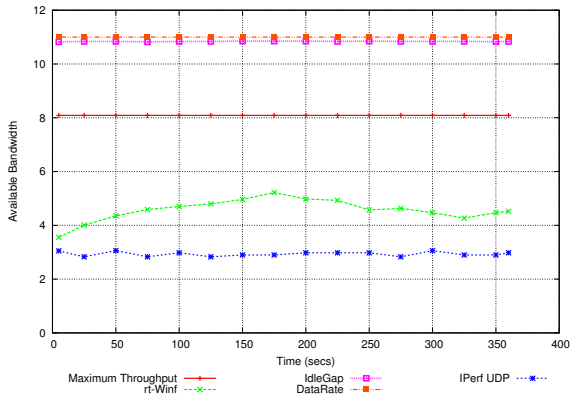


Figure 9. Wireless Mesh Scenario Available Bandwidth

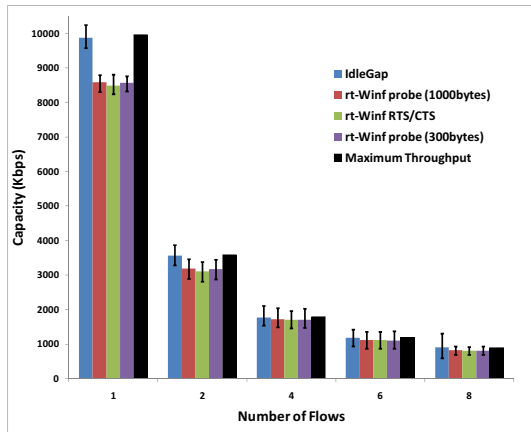


Figure 10. Ns-2 Capacity Results.

In order to compare the values of the emulator with the ones of the simulator, and also to investigate the behavior of rt-Winf with probe packets in a wireless mesh scenario, we also conducted some simulations in the ns-2 simulator [20]. As rt-Winf is based in IdleGap, the simulations also compares those two tools. Each simulation consists of a FTP transfer from a source to a sink, with different simultaneous flows. The maximum throughput is calculated using ns-2 default values and using Equation 5. Figure 10 summarizes the obtained results. Each value is an average of 20 runs lasting 300 seconds of simulated time and nodes are stationary. As observed, IdleGap results are very similar to the the maximal theoretical throughput, as it is using in the calculations the IEEE802.11 Header *DataRate* value. These results validate the ones obtained with the CMU Emulator, since the results for 1 flow in Figure 10 are similar to the ones of Figure 8. It is also possible to conclude that rt-Winf with probe packets (different sizes were used) is also efficiently measuring the capacity, and its values are very similar to the rt-Winf mechanism working with RTS/CTS control packets.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a new mechanism, based in IdleGap, for the measurement of wireless capacity and available bandwidth. It uses information already available in 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. Besides being able to accurately determine the capacity and available bandwidth, rt-Winf does not introduce any changes to the equipment protocol stack. This means that rt-Winf can be supported by any existing wireless equipment without the need to change the Network Interface Card (NIC) drivers. The obtained results, conducted in wireless infrastructure, ad-hoc and wireless mesh networks, show that rt-Winf efficiently performs the desired calculations, providing accurate results without the need to negatively influence the network. rt-Winf can be used in a passive way, measuring the existing traffic of the wireless links, without the need to introduce more traffic in the network.

As future work, we plan to apply this new mechanism to congestion control, scheduling and routing mechanisms.

REFERENCES

- [1] C. D. et al., "What do packet dispersion techniques measure?" *IEEE INFOCOM*, January 2001.
- [2] V. Jacobson, "Pathchar: A tool to infer characteristics of internet paths," *MSRI talk*, April 1997.
- [3] R. K. et al., "Capprobe: A simple and accurate capacity estimation technique." *ACM SIGCOMM*, vol. 34, no. 4, 2004.
- [4] K. L. et al., "Measuring bandwidth," *IEEE INFOCOM*, pp. 235–245, 1999.
- [5] D. A. et al., "Available bandwidth measurement as simple as running wget," *Passive and Active Measurement (PAM) Workshop*, 2006.
- [6] V. J. R. et al., "Pathchirp: Efficient available bandwidth estimation for network paths," *Passive and Active Measurement (PAM) Workshop*, April 2003.
- [7] J. M. et al., "End-to-end available bandwidth: Measurement methodology, dynamics and relation with TCP throughput," *ACM SIGCOMM*, August 2002.
- [8] L.-J. C. et al., "Adhoc probe: Path capacity probing in wireless ad hoc networks," *Wireless Networks*, vol. 15, no. 1, pp. 111–126, 2009.
- [9] H. K. L. et al., "Bandwidth estimation in wireless LANs for multimedia streaming services," *Advances in Multimedia*, vol. 2007, no. 1, pp. 9–9, 2007.
- [10] "IPerf." [Online]. Available: <http://dast.nlanr.net/Projects/Iperf/>
- [11] N. H. et al., "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, August 2003.
- [12] M. L. et al., "Wbest: a bandwidth estimation tool for iee 802.11 wireless networks," *IEEE LCN*, October 2008.
- [13] M. S. Gast, *802.11 Wireless Networks - The Definitive Guide*, 4th ed. O'reilly, 2002.
- [14] *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>
- [15] H. Zimmermann, "OSI reference model- the ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, April 1980.
- [16] P. Brenner, "A technical tutorial on the iee 802.11 protocol," Department of Engineering, BreezeCom Wireless Communications, July 1997. [Online]. Available: http://sss-mag.com/pdf/802_11tut.pdf
- [17] N. B. et al., "Incentive compatible medium access control in wireless networks," *ACM GAMENETS*, 2006.
- [18] "Recommendation ITU-T G.711.0," ITU, 2005. [Online]. Available: <http://www.itu.int/rec/T-REC-G.711/>
- [19] "CMU wireless emulator." [Online]. Available: <http://www.cs.cmu.edu/emulator/>
- [20] "The network simulator - ns-2," 2001. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [21] Željko Ilić et al., "Optimal MAC packet size in wireless LAN," *MIPRO*, 2005.
- [22] V. J. et al., "Congestion avoidance and control," *ACM SIGCOMM*, August 1988.