Field trials of two 802.11 residual bandwidth estimation methods

Martin N. Nielsen
University Graduate Centre (UniK)
Kjeller, Norway
martinnn@unik.no

Knut Øvsthus
Bergen University College
Faculty of Engineering
Bergen, Norway
knut.ovsthus@hib.no

Lars Landmark
University Graduate Centre (UniK)
Kjeller, Norway
larslan@unik.no

***Abstract*—Ad hoc networks have attracted much attention due to their decentralized architecture and potential mobility. The latter promise has sparked research aimed towards routing and quality of service (QoS) admission schemes. The volatile links that are inherent to these networks have however spawned a need for a cross-layering service below IP for local route and QoS decision making, or information dissemination to participating hosts. We have conducted field trials of an active and a passive estimation method implemented in a real wireless driver to estimate the residual bandwidth. Through these tests we show how residual bandwidth estimates can be obtained in an 802.11 wireless network and identify the strengths and weaknesses of their underlying methods. The implemented wireless driver collects per link bandwidth data, local retransmission count and expected transmission time.**

***Index Terms*—Residual bandwidth estimation, field trials, IEEE 802.11, ad hoc networks.**

I. Introduction

Since the advent of the 802.11 technologies and recognition of having self configuring mobile ad hoc networks (MANETs) much attention has been focused at theoretical results and simulation work. Call for a more reliable algorithm for gaining channel access has also emerged since the initial equal competition among the nodes cannot offer any guarantees. Several proposals have been made on how to integrate quality of service (QoS) into ad hoc networks [1, 2]. Most of the proposals for incorporating QoS have either a bandwidth allocation scheme or bandwidth dependent routing. This poses a serious challenge for a shared medium with rate adaptation like an 802.11 based network. In order to allocate bandwidth on demand, a node must know the amount of residual bandwidth available. Knowledge of the traffic surrounding a node can best be observed by the nodes wireless device. These observations in 802.11 networks are complicated by constantly changing link characteristics.

The available bandwidth estimate could either be included in e.g. routing decisions or communicated as cross layer information to the IP layer. The focus of this paper is methods for establishing residual bandwidth estimates, not a solution for usage of these estimates. However, the motivation for cross layer communication is to augment the IP layer with knowledge about the link layer, thus enabling more informed decisions e.g. concerning call admission control (CAC). Cross layer communication of available bandwidth should be implemented as an API to support channel assessment. Such an API is however not standardized.

The basis for our field trials is an implementation made for an 802.11 device. Most analysis and understanding of emerging technologies is often based on simulation results. Network simulators like ns-2, J-sim [3, 4] and alike are popular and often the only tool for assessing new solutions. However it is important to remember that simulations are based on models and simplifications of real physical phenomena. The results have to be interpreted and only a real implementation can verify any solution, since simulations only uncover some characteristics. Also the simplifications made to one part of the model may affect the results in another without being explicitly noticed.

A second challenge concerns what is possible to implement in a real system. Cross-layer architectures are dependent on accessing MAC layer features or even augmenting the MAC. Real systems often impose access restrictions both with respect to the source code and the information available from the physical device itself (e.g. 802.11 system). Thirdly, cross-layer models may introduce assumptions (or constraints) that are not easily met by hardware or operating system architecture, especially on commercially available low cost products.

This paper presents implementations of two differing methods for estimating residual bandwidth and results obtained in real field trials. These methods have been implemented in the Linux driver for Atheros 5212 chipset based wireless devices [5]. The field trials analyzed both a passive listening method [6] and an active method [7, 8] reliant on measuring the transmission time of a MAC layer header. Common for both is that they calculate residual transmission time.

Section 2 presents the two residual bandwidth estimation methods, section 3 will present our field trial setup, with the results presented in section 4. We conclude by discussing the viability of the two methods and how they can complement each other.

II. Obtaining residual bandwidth estimates

There are several terms that are important to keep in mind, when discussing residual bandwidth estimation. The first is the difference between channel share and residual bandwidth. The first relates to the available share a node can attain in competition with other nodes. This can either be larger or smaller than the residual bandwidth. The residual bandwidth is the amount of channel resources unused. If the channel in a network with n neighbouring nodes is saturated one node can still get its 1/n channel share given equal competition. In real scenarios however, the channel is not always saturated.

The second issue relates to the amount of residual bandwidth for a specific modulation rate and frame size. The basis for estimating residual bandwidth is theoretical maximum throughput (TMT) [9]. Throughput in 802.11 networks depends on both modulation rate and frame size. This makes available bandwidth calculation dependent on traffic characteristics since the frame size may vary greatly between e.g. ftp flows compared to voice. Small framed flows introduce a lot of transmission overhead in the form of back off, physical headers and link layer acknowledgments.

Transmission time for a frame follows the following formulas, and is dependent on the protection used.

$$T_d^{CSMA/CA} = T_{DIFS} + T_{backoff} + T_{DATA} + T_{SIFS} + T_{ACK} \quad (1)$$

$$T_d^{CTS-to-self} = T_{DIFS} + T_{backoff} + T_{CTS} + (2 \times T_{SIFS}) + T_{DATA} + T_{ACK} \quad (2)$$

$$T_d^{RTS/CTS} = T_{DIFS} + T_{backoff} + T_{RTS} + T_{CTS} + (3 \times T_{SIFS}) + T_{DATA} + T_{ACK} \quad (3)$$

T_d^{CSMA} is the transmission time for unprotected frames, $T_d^{CTS\text{-}to\text{-}self}$ is for CTS only protected frames, and $T_d^{RTS/CTS}$ is for fully RTS/CTS protected frames. By looking at Figure 1 one can see that T_{DIFS} is the initial waiting period before sending a frame. $T_{backoff}$ is the random waiting period drawn to allow uniform channel access probabilities for competing nodes while minimizing collision probability. The optional $T_{RTS/CTS}$ protects data frames from collisions, while the T_{SIFS} allows the radio to change from Tx state to Rx and vice versa. The T_{ACK} is the time notification of Tx-success for unicast frames takes.

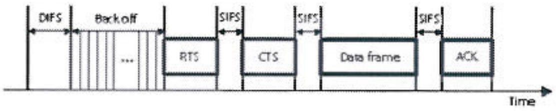

Figure 1 – A fully protected frame exchange in 802.11.

The TMT is calculated on the basis of the number of data frames that can be transmitted within an interval. T_{DATA} is the only part of the transmission time that is affected when modulation rates change. The other factors remain constant for a given physical medium (802.11a/b/g) and preamble type. The two methods for estimating residual bandwidth are independent of rate considerations for an initial estimate as they give a percentage approximation. A calculation of residual bandwidth is then made based on traffic characteristics (modulation rate and frame size). During analysis of the field trials the TMT will be used as a reference, not only for calculating the residual bandwidth for a given rate and frame size, but also for evaluating if the results presented by the two methods are fair compared to the real channel conditions.

The passive and the active methods rely on calculating an average time a frame should occupy the local device when attempting to traverse the wireless channel. As seen in Figure 1 and presented by equation 1, 2 and 3 the time to traverse the channel has two variable parts, the data frame and the back off period. For pre-calculating the T_d, the back off period is an average of the random back off expected. T_{DATA} is calculated on the basis of the frame size and modulation. We continue by showing how the passive method uses the T_d to calculate the residual bandwidth, with the active method following.

A. Passive bandwidth estimation

The passive estimation method is interval based, and calculates T_d for every sent and received frame. For the received frames the $T_{backoff}$ and the T_{DIFS} are omitted from eq. 1-3. This is considered idle time by the receiving hosts. The T_d values calculated on a per frame basis are summed. This gives us the amount of time the channel is occupied during the interval. Residual bandwidth is then estimated from the remaining idle time by using information about used modulation and averaged frame sizes per peer. This information is readily available through the history kept.

The T_d from the TMT calculation assumes that no retransmissions occur. The passive estimation method also calculates values for any locally occurring retransmissions by utilizing eq. 1-3 and increasing the back off interval to a predefined average value and summing all the retransmission attempts. The back off interval doubles for every retransmission attempt until it reaches a 1023 slots. By having the passive estimation code at the MAC layer statistics such as retransmission attempts are readily available.

B. Active bandwidth estimation

The T_d calculation is equally important for the active estimation method. However, the T_d is used as a reference point to how the channel should behave when unoccupied. The wireless device halts the back off countdown when the channel is sensed busy by transmissions from competing hosts and the T_d can therefore be longer in reality. The active estimation method observes the channel for an observation period. This period is then divided into discrete intervals ($\varDelta T$). One probe, in our case a frame only containing a 802.11 Ethernet header, is scheduled at a random point within a $\varDelta T$ interval. And the actual time from placing the frame onto the wireless device till it notifies the driver that an acknowledgment has been received is measured. By comparing T_d to the measured value, T_m, one can calculate the time used to defer from the channel, thereby estimating the percentile of channel usage. The channel utilization estimate is calculated by the following expression:

$$\hat{u}_{c,i} = \frac{\sum_{j=1}^{j=K_{c,i,j}} (T_m^{i,j} - T_d^{i,j})}{K_i \Delta T_{estimated,i}} \tag{4}$$

$\hat{U}_{c,j}$ gives the channel utilization estimate for a given probe modulation rate, j. We do not set any specific rate for the probes in order to lessen the burden on the channel by these. The MAC rate controller decides the Tx rate. Superscript *i* denotes the i'th probe having a transmission time longer than T_d. Results that gave a T_m shorter than T_d were not included in the summation. K_i is the number of probes sent at a specific rate, and $\varDelta T_{estimated}$ is the size of the interval the probes were scheduled inside. This value is estimated because T_m can be much larger than $\varDelta T$ which would imply queuing of probes. We do not allow this for two reasons. Firstly, because there is no way of knowing when a frame is de-queued by the wireless device thereby introducing queuing time into the estimate, our implementation avoided this problem. Secondly, this reduces the load on the channel as cross traffic increases. By not scheduling new probes until the queued probe is sent, the probe load is adaptive. Because the probes are sent at differing rates, $\hat{U}_{c,j}$ is averaged between the rates by a weight function. The weight being the number of probes sent at each rate.

III. Field trials

Several field trials were conducted to analyse the implemented bandwidth extensions made to the driver. To ensure that these were not influenced by neighbouring 2,4GHz sources the field trials were performed on a parking lot deep inside the Norwegian woods. The test bed consisted of five laptops equipped with regular 802.11b wireless devices. A sixth laptop contained the Atheros based device and the experimental driver (estimator node in Figure 2 and 3). The implemented extensions are not dependent on having special features evident in the surrounding nodes. The tests were conducted by having the regular nodes conversing and letting the monitoring node assess the channel both passively and actively. The conversation between the hosts was enabled by using mgen [10] and logging the transmission and reception. The logging monitored the goodput not the channel load.

	Scenario 1	Scenario 2	Scenario 3
Topology	One hop	One hop	Chain
Protection	RTS/CTS	RTS/CTS	RTS/CTS
Dev. Rate	11Mbit	11Mbit	1Mbit
Traffic pattern	Periodic	Periodic	Periodic
Pkts/s	50, 150, 300	45, 99, 210, 600	50, 150, 300
Frame size	256, 756, 1456	256, 756, 1456	256, 756, 1456
Tx peers	B	B, C, D	B
Rx peers	A	A, E	A

Table 1 - Host setup for mgen.

Three main scenarios were arranged for the trials. To ascertain the accuracy of the bandwidth estimates, the channel load was varied by changing both the packet rate and frame size. Scenario 1 (ref. Table 1) was conducted in a one hop topology to ascertain if the residual bandwidth estimates were accurate when only one flow occupied the channel. Scheduling of frames on the medium, collision avoidance and decoding ability have different characteristics when the channel is contention free compared to when other host compete for channel access. Therefore we also devised a second scenario, scenario 2 (ref. Table 1), where three competing hosts contended for channel access. Scenario 3 (Figure 3) was conducted to decide if it was possible to sense the two-hop traffic node B was sending to node A. Given that RTS/CTS was turned on this should be possible for the active method since the Atheros device should defer from the channel if it hears the CTS replies from node A. The build-up of the passive estimation driver code is done in such a manner that it discards control frames; it should therefore not be possible to perform such sensing with it.

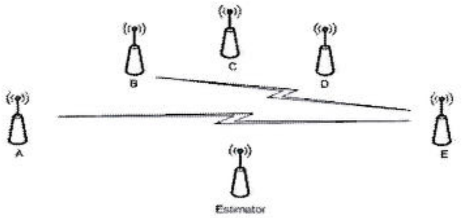

Figure 2 – One hop topology, with varying number of conversing hosts. Estimator is sending probes to node A.

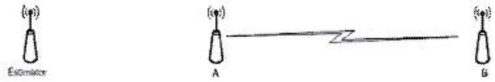

Figure 3 – Scenario 3, chain topology, with node B sending traffic towards node A, estimator node outside of node Bs range. Estimator is sending probes to node A.

The passive estimation engine did not need any additional setup for the field trials except for enabling promiscuous mode for the device itself. The active however has several variables that have to be set. The first is the ΔT which controls the spacing between the probes and the effective channel load given a large enough DCF share. ΔT was set to so that the channel occupancy reached 40% on an idle channel. The second parameter was the number of probes. This was set to 500, however the method allows for calculation of bandwidth by using only a subset of these. This flexibility gave an opportunity to calculate an optimum number for use as a recommendation which was found to be 200 probes, this coincides with the simulation results in [7]. The third parameter is the MAC address of the host targeted for the estimate. Host A was selected for this.

IV. Results

To analyse the results special attention was paid to how accurately the active and passive methods estimated the channel conditions, as well as the effect on the regular traffic by the probes generated by the active method. The accuracy was measured by pre-calculating a channel load estimate given a frame size, packet rate and modulation. This is shown as a straight line in the figures and is calculated as the theoretical load a specific traffic pattern should exert. The passive and active estimates were obtained independently. The passively obtained results are based on stabilized values that were collected after enough intervals had passed to make them representative. Then 10 probing sessions followed. These are based on measurements on a much smaller timescale and will therefore vary more.

The results accumulated for scenario 1 show that the active method does have an issue with estimating the channel for small frames (in μs, not bytes). This happens due to the fact that small frames are more easily multiplexed with the probes and are not sensed as well. The active estimation code includes a random scheduler for the probes but this does not seem sufficient to sense periodic traffic from one host. The error increases as the channel load is raised. For the bigger frames this is issue is not as prominent. The active results show a spread in the measurements. The observation period is much smaller than for the passive method. This means that cross traffic variations due to channel access contention will be more visible. Any additional back off performed by the peers due to frame losses is therefore more prominent in the estimates. As the passive method bases its estimates on longer intervals the variations will be minute, especially when the packet generation rate, modulation and frame size is kept constant. The passive method estimates the channel very well.

The addition of more hosts on the channel in scenario 2 in order to add randomness to the periodic pattern, by virtue of carrier sensing channel deferral, improved the estimates for the same total channel load. As the tests were conducted at 11Mbit, the largest frames showed most promise. Tests done early in the development conducted with rates at 5,5Mbit did show that considerable accuracy was achieved already at medium sized frames. This dependency on frame size was not evident in the simulations done in [7] and can be attributed to both the simplifications made in network simulators and the alterations we had to make to the method due to inability to implement [7] unaltered in the real driver.

A further issue concerns the inability to correctly assess the channel for high channel utilization. The reasons for this are threefold. Firstly, when the channel utilization becomes this high there will be more evidence of the capture effect. As noted in [11] this could be up to 25%.

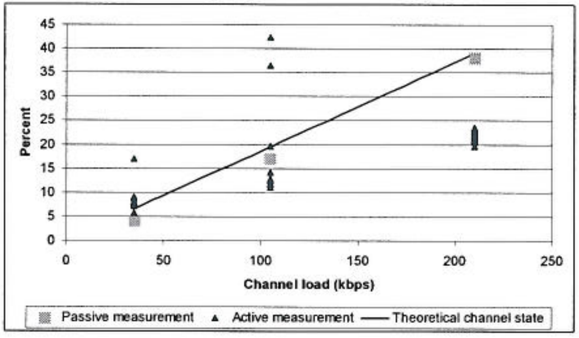

Figure 4 - Scenario 1, medium frame size.

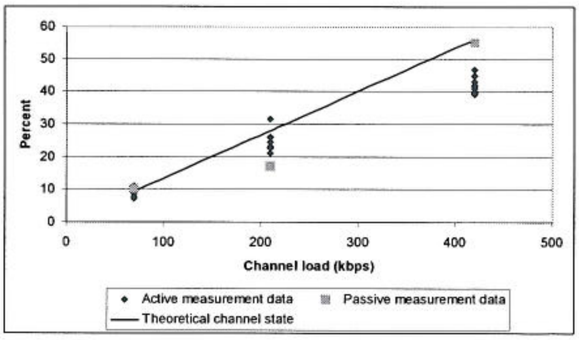

Figure 5 - Scenario 1, large frame size.

Secondly because ΔT is small, then the cross traffic will experience competition for channel resources. This will reduce the cross traffic and will be reflected in the active probing estimates. Thirdly, the alteration of the original method meant that we stretch the ΔT for probes that take longer than ΔT. We also keep an inter-probe gap. This will lower the maximum achievable estimate, because the stretching of ΔT including the gap will never allow ΔT to be 100% utilized. The passive method underestimates the channel to some degree as the cross traffic load increases. The active method does the same to a greater extent, but also affects cross traffic. This is seen for the high loads depicted in Figure 6 and 7 where passive listening during probing reflects a lowered cross traffic. The third scenario which was tested to see if two-hop traffic could be sensed showed that it was indeed possible. However, setting up this experiment proved to be difficult because we wanted to test the ability to sense the two-hop traffic indirectly by probing the closest hop (node A), not by deferral

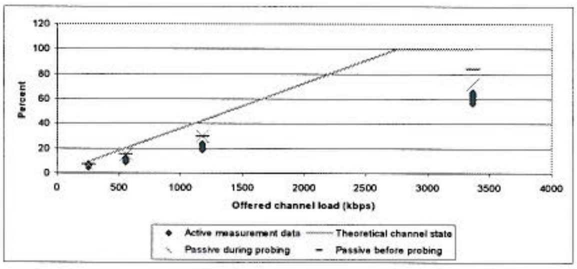

Figure 6 - Scenario 2, medium frame size.

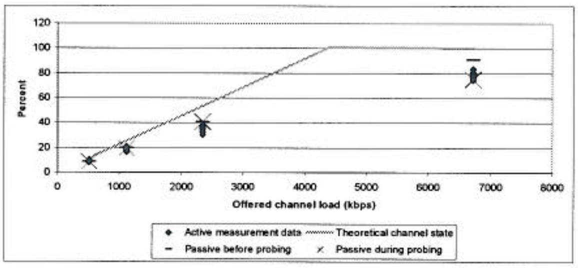

Figure 7 - Scenario 2, large frame size.

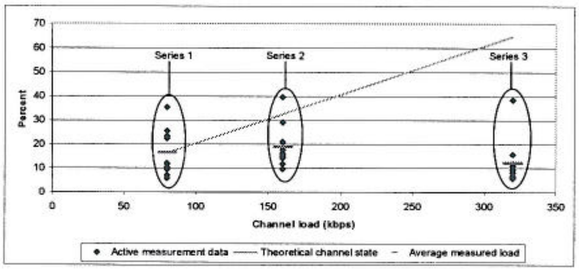

Figure 8 - Scenario 3, active estimation for two hop case.

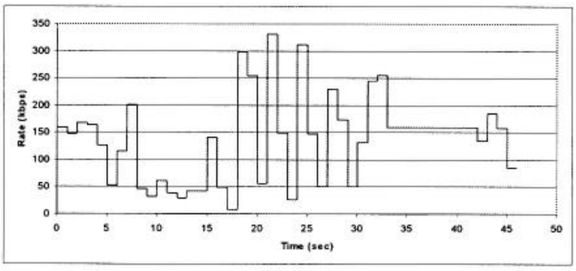

Figure 9 - Scenario 3 mgen throughput plot for Series 2, under an offered load of 160kbit/s.

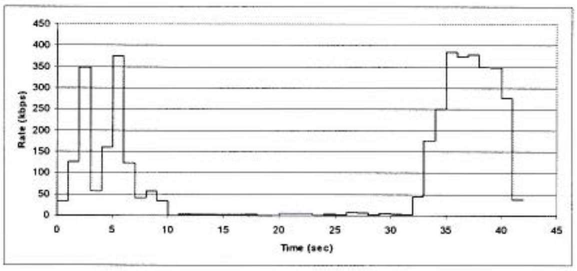

Figure 10 - Scenario 3 mgen throughput plot for Series 3, under an offered load of 160kbit/s.

due to carrier sensing caused by node B. Since the interference range is much larger than the transmission range, this test was conducted at 1Mbit at the rim of the transmission range for all hops. This resulted in non optimal transmission quality and the throughput of these tests showed a variation even when no probing traffic was present. As a consequence the actively sensed traffic varied much more than in the other tests. For the two-hop traffic the passive method gave no reading. The active method gave off-putting results when looking at Figure 8; however the results gave a good correlation between the measured results and the reported throughput of mgen. As the offered traffic load increased, the added probe traffic reduced the traffic flow (i.e. the mgen traffic). The effect of the probe traffic is evident from the mgen throughput plot, as seen in Figure 9 and 10. Figure 10 shows that the active probing resulted in greatly reduced traffic from node B, and that the channel was left almost idle when the offered load was 320kbit/s. This means that reduced throughput accompanied by active probing has a detrimental effect. When the offered load was 160kbit/s the throughput was less affected. As the traffic load increases, less resources are left for each node. Our active probes were small in size and therefore also more likely to experience a successfully transmission compared to larger frames. However the results were inconclusive to ascertain the relative contribution between these two throughput reducing factors, the extra channel load introduced by the probes and the difference in packet size.

V. Discussion

The results showed that bandwidth estimation based on passive listening has its limitations, as does estimation based on the active method. When estimating passively miscalculation in the channel utilization estimates occurs due to position independence between nodes. This was observed in scenario 3. The traffic heard at one node did not reflect the peers channel state. The channel occupancy is closely related to the channel state at both the sender and the receiver. The channel has to be free at both ends to enable communications. In active probing sessions the observed T_m is in part reliant on local conditions and in part on the conditions at the receiving peer due to reception of an ACK. As a result, the observed relationship includes the peer channel states as these are taken into account. A further concern is for silent neighbours when the passive method has little information to infer link capacity.

The scenarios we chose, as well as the location, did not include cross-traffic coded at too high modulation so that it would pass undetected by the MAC sub-layer of 802.11. The channel was also not affected by high bit error rates. This meant that we did not investigate sensing of channel deferral with the active method in the one hop scenario for these two cases. The active method should sense concurrent frames that cannot be decoded as the physical carrier sensing must be influenced by the transmission and forcing channel deferral.

The results confirm that the probes do have an impact on the other flows, even the two-hop. For scenario one and two the impact was relative to the channel share. With the second scenario having more nodes competing for channel access the increased load caused by the probes was dampened as the DCF spread the contention probability. Such impact is often common for measurements in general. The transmissions of a probe will undoubtedly influence other traffic sources and even the local Tx-queue. The active measurement has an effect on the observed channel by introducing latency and jitter, and as frames are introduced to the environment resources are stolen. For the passive method there is a lot of processing done in kernel. This can take valuable CPU resources and introduce process scheduling latency and jitter. Depending on the size of the introduced interference either active or passive this can have a very intrusive effect.

VI. Conclusion

We have implemented an active bandwidth estimation technique and tested it with the passive in real field trials. These are running in parallel within the same driver. The results show that the passive method has good accuracy when cross-traffic frames are easily decoded. The active is better at sensing the channel when the carrier sensing mechanism blocks channel access. It does however affect the channel, this is negative especially under high traffic loads.

References

1. Ahn, G.-S., et al., *SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks*, in *IEEE INFOCOM*. 2002: New York.
2. Perkins, C.E. and E.M. Belding-Royer, *Quality of Service for Ad hoc On-Demand Distance Vector Routing, Work in progress*. 2003.
3. University of California at Berkeley, C., *Network simulator 2 (ns-2)*, http://nsnam.isi.edu/nsnam/index.php/Main_Page.
4. Miller, J., et al. *JSIM: A Java-Based Simulation and Animation Environment*. in *30th Annual Simulation Symposium*. 1997. Atlanta, Georgia.
5. Leffler, S., M. Renzmann, and G. Chesson, *Multiband Atheros Driver for WiFi* http://sourceforge.net/projects/madwifi/.
6. Déziel, M. and L. Lamont, *Implementation of an IEEE 802.11 Link Available Bandwidth Algorithm to allow Cross-Layering*. 2005, CRC Canada.
7. Li, F.Y., et al. *Estimating Residual Bandwidth in 802.11-based Ad Hoc Networks: An Empirical Approach*. in *Proceedings of the 7th International Symposium on Wireless Peronal Multimedia Communications (WPMC'04)*. 2004. Abano Terme, Italy.
8. Nielsen, M. *Demo of residual bandwidth estimation in an 802.11 ad hoc networks*. in *REALMAN'05*. 2005. Santorini, Greece: IEEE.
9. Jun, J., P. Peddabachagari, and M. Sichitiu. *Theoretical Maximum Throughput of IEEE 802.11 and its Applications*. in *IEEE Int. Symposium on Network Computing and Applications (NCA)*. April 2003.
10. Naval Research Lab, *The Multi-Generator Toolset (mgen)*: http://pf.itd.nrl.navy.mil/mgen/.
11. Kochut, A., et al. *Sniffing out the correct physical capture model in 802.11b*. in *12th IEEE International Conference on Network Protocols (INCP'04)*. 2004. Berlin, Germany.