An UDP Protocol Booster for Multimedia Communications over 802.11 Wireless LAN

*Abram Schoutteet[1], Jeroen Hoebeke[2], Tom Van
Leeuwen[3], Ingrid Moerman[4], Bart Dhoedt[5], Piet
Demeester[6], Nico Janssens[7], Pierre Verbaeten[8]
Department of Information Technology (INTEC),
Gent University-IBBT-IMEC, Gaston
Crommenlaan 8 bus 201,
9050 Gent, Belgium,
Tel: 00 32 9 331 49 00
[7,8]K.U.Leuven
Departement Computerwetenschappen
Celestijnenlaan 200A
B-3001 Leuven, Belgium
E-mail: [1]Abram.Schoutteet@intec.Ugent.be
[2]Jeroen.Hoebeke@intec.Ugent.be
[3]Tom.VanLeeuwen@intec.Ugent.be
[4]Ingrid.Moerman@intec.Ugent.be
[5]Bart.Dhoedt@intec.Ugent.be
[6]Piet.Demeester@intec.Ugent.be
[7]Nico.Janssens@cs.kuleuven.ac.be
[8]Pierre.Verbaeten@cs.kuleuven.ac.be

Abstract: The standard 802.11 WLAN protocol performs well for data communications. However, in contrast to data communications real-time audio and video communications require strict bandwidth, delay and jitter guarantees. Because of the typically bursty error characteristics of a wireless channel the 802.11 WLAN protocol can not always fulfil those requirements. In this paper we present the design of an UDP protocol booster based on negative selective acknowledgements which takes into account the playback deadline of real-time packets and tries to predict packet loss in order to reduce the overall delay. A test bed implementation demonstrates the protocol booster considerably improves the overall performance in the presence of long error bursts on the wireless channel.

1. Introduction

IEEE 802.11 Wireless LAN (WLAN) is currently one of the most widely used wireless technologies. Its main characteristics are simplicity, flexibility and cost effectiveness. 802.11 WLAN provides people with a ubiquitous environment for communication within their offices, homes, campuses, etc. At the same time the use of multimedia applications is increasing exponentially. People are expecting high quality video and audio even while moving in the office or travelling around campuses.

In contrast to data communications, which are often very sensitive to packet loss, audio and video communications also require strict bandwidth, delay and jitter guarantees. This poses many challenges as wireless links suffer from a high loss rate, bursts of frame loss, large packet delay and jitter. Furthermore these characteristics are not constant and vary over time and place.

The current 802.11 WLAN protocol [1,2] uses a Stop and Wait Automatic Repeat Request Protocol (SW-ARQ), which is a simple and very efficient technique that performs well for data communications, but can introduce large packet delays. It requires a receiver to send an Acknowledgement (ACK) to the sender for each correctly received MAC frame. Frames that fail the Cyclic Redundancy check at the receiver are simply ignored. If no ACK is received shortly after transmission, the sender retransmits the packet after a random delay (which increases with the first few retransmissions of the same packet), repeating this process as long as needed, until an ACK is received or until it reaches the maximum retransmission threshold (e.g. 7), at which point the sender gives up. This behaviour introduces large packet delays if long error bursts occur and the maximum retransmission threshold has a high value.

Every real-time streaming packet has a playback deadline and becomes useless if received too late at its destination due to multiple retransmissions. The 802.11 WLAN protocol has no knowledge of this deadline information and will unnecessarily retransmit packets.

There has been done considerable research on the topic of quality of service (QoS) support for IEEE 802.11 WLAN. Some techniques make use of service differentiation based schemes, for example the EDCF [3] and HCF [3, 4] schemes that are used in the forthcoming 802.11e standard. Other techniques make use of error control schemes, for example SE-ARQ [5], GBN-ARQ [5] or forward error correction. Those techniques all have in common that they adapt the standard 802.11 protocol itself.

In the approach presented in this paper we tackle these problems by reducing the maximum 802.11 retransmission threshold to zero or a low value (to eliminate the delay), and wrapping an extra retransmission protocol around the standard 802.11 WLAN protocol. This protocol not only does not retransmit packets that will miss their playback deadline but also makes use of selective negative acknowledgements (SNACKs) to reduce the amount of redundant traffic and extra delay that would otherwise be introduced by the error control protocol over the wireless link.

We base our method to improve the performance of real-time multimedia communication over 802.11 WLAN on the concept of protocol boosters. Protocol boosters are modules that improve the performance of the original protocol [6, 7]. Because they do this in a transparent way, i.e. without modifying the end-to-end messages of the original protocol, transmissions can still continue even if the booster fails, although with reduced performance. A booster can be inserted in the network at any place (e.g. at an access router or a mobile host) and can add, remove or delay protocol messages. The main advantages of protocol boosters are their ease of deployment (there is no need to adapt the original protocol) their adaptability to make them suit the specific needs of the network environment and their robustness. Since most multimedia communications use UDP/IP as their transport protocol because it introduces very little overhead (e.g. compared to TCP/IP) we focus on this protocol.

The remainder of this paper is structured as follows. In the section 2 we present the booster, the components it is composed of and the improvements it makes in the transmission of multimedia communication streams. In section 3 we present results of our experiments executed on the Click Modular Router platform [8].

2. UDP booster protocol

The booster we describe consists typically of two booster elements. One element is placed before the wireless link (for example at an access router) and one behind the link (for example at a wireless host). In the protocol stack the booster protocol is situated between the MAC layer and the network layer, but makes use of information from the network layer, transport layer and even from the application layer.

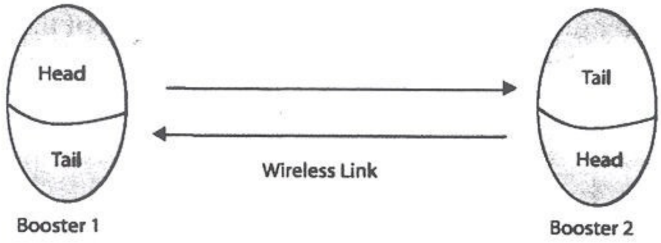

Figure 1 – Functional structure of the booster

Each booster element is split into a booster head and a booster tail (figure 1). The head of the first booster element stores a copy of all arriving packets and the time of their arrival at the booster element in a circular FIFO buffer. It then adds additional booster specific information to the options field of the IP header of the packet. This information consists of a flag indicating whether the packet is a regular or a retransmitted packet and a sequence number that is incremented with each new arriving packet. It then passes the packets on to the MAC layer. We make the assumption that packets arrive at the second booster element in the same order they were handled at the first booster element. The tail of the second booster element removes the added booster options from the IP header of the packets. If the tail notices one or more packet were lost over the wireless link (based on the range of sequence numbers it did not receive) it sends a SNACK to the first booster element asking for the lost packets (see figure 2). A SNACK has priority over all other traffic that needs to be sent over the wireless link.

2.1 Retransmissions based on selective negative acknowledgements

When the head of the first booster element receives a SNACK it will retransmit only those packets that will actually arrive at their destination before the playback deadline. The information needed by the first booster element to achieve this is included by the second booster element in a Maximum Delay Field in the SNACK. This field contains the maximum delay packets may sustain over the wireless link to still be capable of arriving at their destination before their playback deadline, taking into account the round trip time of the link. This assumes an estimation of the round trip time is made at the tail of our second booster element. We do this by calculating its moving average using measurements taken during earlier retransmissions. After sending the SNACK the second booster element resumes the processing of the packet stream.

It is of course possible that the SNACK, some or all packets that are retransmitted over the wireless channel are corrupted. To overcome these packet losses we introduce two extra error recovery mechanisms.

A first mechanism tries to overcome the loss of a SNACK or the loss of all retransmissions by introducing a retransmission timeout trigger. This timer starts when a SNACK is transmitted and stops when all retransmissions have correctly arrived. When the trigger fires the SNACK is sent again and the timer is restarted (see figure 3). This continues until all retransmissions have arrived, a maximum SNACK retransmission threshold is reached or until the packets will not be retransmitted at the first booster element because they can not arrive at their destination before their playback threshold, at which point the booster gives up. The expiration time (Texpire) of the timer is calculated using the estimated round trip time and the standard deviation on that estimation.

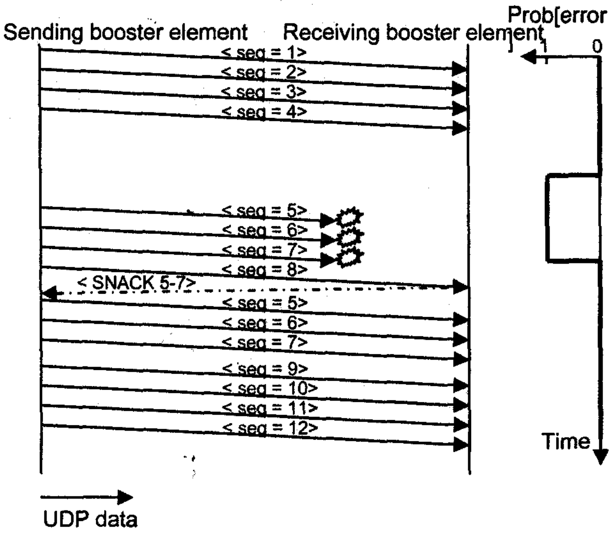

Figure 2 – retransmissions based on selective negative acknowledgements

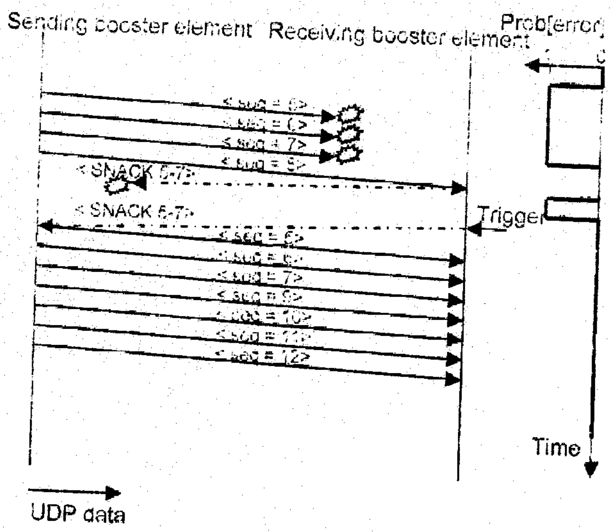

Figure 3 – retransmission of a lost SNACK using the retransmission timeout trigger

A second mechanism tries to overcome the loss of several packets within the retransmission stream (when at least one packet arrives). When the booster tail detects a gap in the sequence numbering of the retransmitted packets, a new SNACK is created and sent to the first booster element asking for the lost packets (see figure 4). To detect packet loss at the end of the retransmitted packet stream the retransmission timeout trigger used in the first mechanism is reinitialized and restarted when the first packet of the retransmission stream arrives. It is reinitialized with a timeout value equal to the estimation of the time it will take the remaining packets of the retransmission stream to arrive at the second booster element. When the trigger fires a SNACK is created for the remainder of the packets and the timer is reinitialized with its original value (Texpire) and restarted (see figure 5).

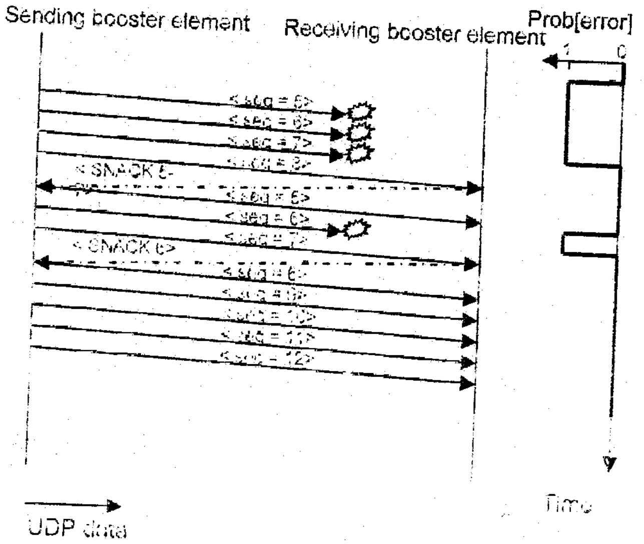

Figure 4 – retransmission of a lost retransmitted packet

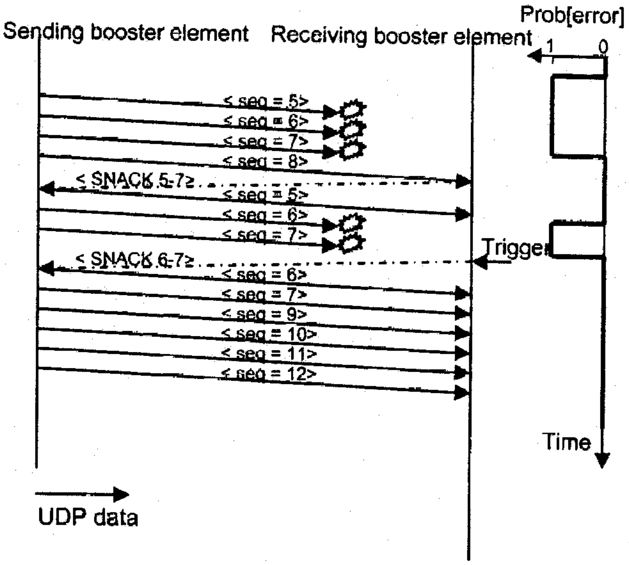

Figure 5 - retransmission of lost retransmitted packets using the retransmission timeout trigger

Since it may occur that a SNACK is sent to recover a series of corrupted packets before all packets asked for by a previous SNACK have arrived, a single retransmission timeout trigger will not suffice. At least a second retransmission timeout trigger needs to be added to the protocol.

2.2 The regular traffic timeout trigger

A big disadvantage of using SNACKs is that the tail of the second booster does not know packets were lost over the wireless link until it receives a new packet and can verify its sequence number. For example: the tail of the second booster will not know packets with sequence numbers n, n+1, …, n+k were lost over the wireless link until it receives a packet with sequence number n+k+1.

This disadvantage can be partially overcome if the communication stream passing through the booster elements is regular. With a regular packet stream we mean that the variance on the packet inter-arrival time or the variance on the time between bursts of packets (if the traffic is bursty, e.g. with most video streams) is relatively small.
If this is the case, we should be able to predict when the next packet or burst of packets should arrive. If the booster tail of the second booster does not receive any packets for a period larger than the time we estimated to exist between the packets or bursts, it is highly likely some packet loss occurred on the link.

To make use of this knowledge a regular traffic timeout trigger is created that restarts with each arrival of a packet (or burst if the stream is burst based). The trigger fires after a fraction (for example 160%) of the time we estimated to exist between the packets (or bursts) has passed, and sends a SNACK asking for all packets that are deemed to be lost. The timer restarts after it sent the SNACK and may expire several more times, the first "maximum SNACK retransmission threshold" times (see above) after a time Texpire and subsequently after a larger period, before the retransmitted packets or a new packet arrives (figure 6). When the head of the first booster element receives such a SNACK it retransmits all packets with a sequence number greater than the lowest sequence number mentioned in the SNACK that fulfil the maximum delay constraint.

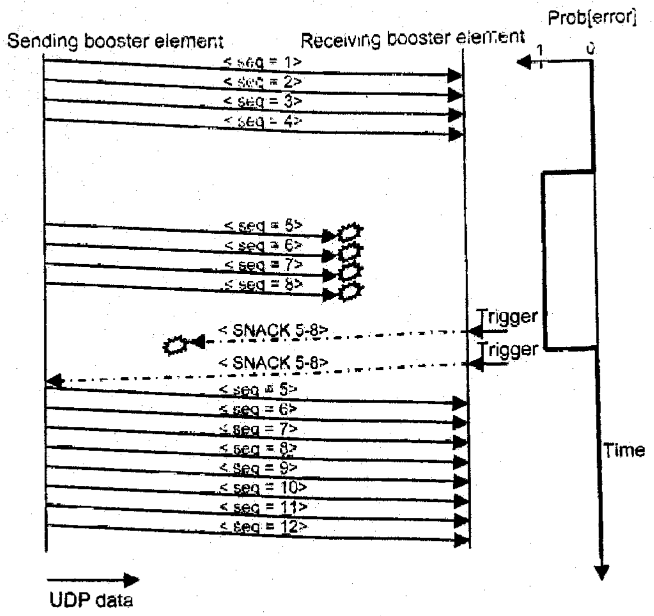

Figure 6 – anticipating packet loss using the regular traffic timeout trigger

2.3 Atomic frames, packet reordering and fairness

If the video stream is burst based and the player at the client side can not handle partially received frames (which can arrive at the client when a frame can not be retransmitted in its whole within the maximum delay period) we can further optimize the protocol booster by not retransmitting such partial bursts. To be able to do this the head of the first booster element needs to buffer the time of arrival of each new burst.

With this booster protocol we do not preserve the sequence order of the packets that are sent over the wireless link. If this proves to be necessary packets will have to be buffered and re-sequenced before they can be delivered to the application. A less complex solution could be to let the head of the first booster element not only retransmit those packets asked for by a SNACK but also all the following packets.

If we study the situation where a single booster head serves more than one booster tail and the error rate on the wireless link between the booster head and one of the booster tails is higher than that of the other links, unfairness can be introduced.

As a direct consequence of the retransmissions the throughput on the error prone link will be higher with the booster pair than it would have been without it. This extra bandwidth is taken from all links. To rectify this situation we can introduce Round Robin scheduling. Packets and retransmissions from each distinct wireless link are stored in a separate queue and all those queues are scheduled following the Round Robin principle.

3. Click Router implementation results

We evaluated our UDP protocol booster on a Click Router test-bed (figure 7). A video server was installed on the sender and a video client on the receiver. Three Click router PC's connected the server with the client; the outer two were running the booster protocol software and the central PC a model of 802.11 WLAN on top of an error model that emulated packet loss over the wireless link.

The parameters used for the 802.11 WLAN model can be found in table 1.

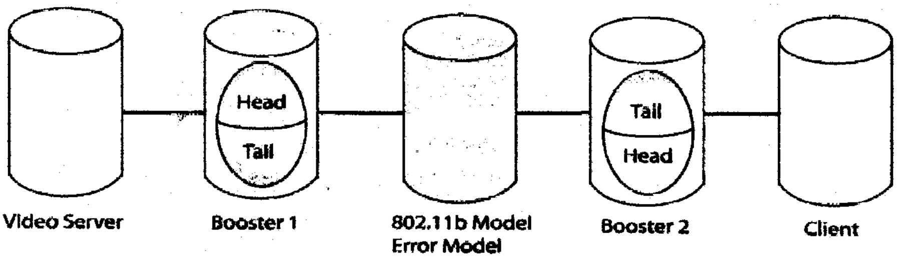

Figure 7 – Click router test-bed

The video stream that was used for the experiments is a 2 Mbps MPEG2 Quicktime movie with a GOP (group of pictures) of 15. The value of the maximum delay over the wireless link was chosen to be 120 ms.

Bandwidth	11 Mbps
SIFS	10 µs
DIFS	50 µs
Preamble length	192 µs
ACK	14 Bytes
Aslot	20 µs
Cwmin	31 slots
Cwmax	1023 slots

Table 1 – parameters used in the 802.11 WLAN model

The error model we used is a two state Markov model (the Gilbert model [9], shown in figure 8). α stands for the chance a packet will not be corrupted over the wireless link if the previous packet was not corrupted and β for the chance a packet will be corrupted over the wireless link if the previous packet was corrupted.

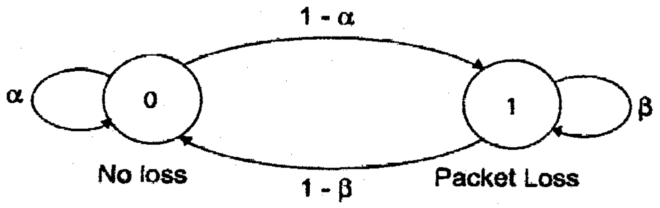

Figure 8 – Gilbert Model for the wireless channel

Table 2 and the last column of table 3 show the packet loss of the video stream as a function of α and β for different configurations of the UDP booster Protocol.

With these experiments the maximum retransmission threshold of the 802.11 WLAN protocol was set to 0 and bursts that could not be retransmitted as a whole within the maximum delay period were not retransmitted. From the experiments it becomes clear that both a higher value for the maximum SNACK retransmission threshold and the use of the regular traffic timeout trigger improve the performance of the booster protocol.

Error Model		**maximum SNACK retransmission threshold = 0 and no regular traffic timeout trigger**	**maximum SNACK retransmission threshold = 3 and no regular traffic timeout trigger**	**maximum SNACK retransmission threshold = 0 with regular traffic timeout trigger**
α	β	Packet Loss (%)	Packet Loss (%)	Packet Loss (%)
0,965	0,60	0,850	0,208	0,825
0,965	0,75	1,454	0,538	1,624
0,965	0,82	2,551	1,175	2,521
0,965	0,90	7,700	6,119	6,649
0,975	0,60	0,460	0,096	0,380
0,975	0,75	0,862	0,342	0,894
0,975	0,82	1,262	0,827	1,282
0,975	0,90	4,434	3,694	4,690
0,985	0,60	0,156	0,059	0,144
0,985	0,75	0,259	0,149	0,290
0,985	0,82	0,562	0,460	0,557
0,985	0,90	2,239	1,826	1,902
0,995	0,60	0,020	0,010	0,043
0,995	0,75	0,020	0,020	0,053
0,995	0,82	0,183	0,093	0,089
0,995	0,90	0,645	0,361	0,255

Table 2 – Packet loss in function of the α and β, parameters of the Gilbert model, for different configurations of the UDP protocol booster

Table 3 shows the packet loss of the video stream in function of α and β. The first two columns show results from the same experiment, executed without the UDP protocol booster and with the maximum retransmission threshold of the 802.11 protocol set to 7. In the first column we do not take the playback deadline into account when measuring the packet loss. In the second column we discard packets that arrived too late at their destination.

Error Model		802.11 WLAN protocol without booster protocol and no playback deadline	802.11 WLAN protocol without booster protocol, with the playback deadline	**Booster protocol: maximum SNACK retransmission threshold = 3 with regular traffic timeout trigger**
α	β	Packet Loss (%)	Packet Loss (%)	Packet Loss (%)
0,965	0,60	0	0	0,208
0,965	0,75	0,058	1,500	0,634
0,965	0,82	0,283	6,887	1,104
0,965	0,90	1,362	33,786	5,163
0,975	0,60	0	0	0,056
0,975	0,75	0,023	0,126	0,238
0,975	0,82	0,151	2,388	0,541
0,975	0,90	1,018	20,98	3,230
0,985	0,60	0,005	0,005	0,043
0,985	0,75	0,005	0,005	0,170
0,985	0,82	0,087	1,393	0,269
0,985	0,90	0,595	4,812	1,418
0,995	0,60	0	0	0
0,995	0,75	0,005	0,005	0,020
0,995	0,82	0,063	0,424	0,010
0,995	0,90	0,197	1,367	0,283

Table 3 – Packet loss in function of the α and β, parameters of the Gilbert model, with and without the UDP booster protocol and playback deadline

When a playback deadline has to be respected and long errors burst (high β) occur on the wireless link the packet loss is greatly reduced (up to a factor 6) with the aid of the UDP booster protocol. If the errors are more spread out through time (low β) the packets will only sustain a small amount of delay when using the 802.11 WLAN protocol (with 7 retransmissions) and it will outperform the protocol booster.

4. Conclusions

We have developed an UDP Protocol Booster based on negative selective acknowledgements. It recovers only those packets that will arrive at their destination before their playback deadline. In addition to this it tries to predict packet loss by analyzing the temporal structure of the packet stream. Using these mechanisms it limits the delay real-time streaming packets may sustain over the wireless link. The performance of the protocol has been validated by Click Router implementations in a test network. It has been demonstrated that the protocol booster considerably improves the performance of real-time UDP-streams, in particular in case of bursty errors over the wireless link.

Our future work includes a performance study of the protocol booster when combining it with a higher maximum retransmission threshold of the 802.11 protocol (e.g. 2) while streaming video over an error-prone wireless link. We will further try to improve and extend the capabilities of the UDP protocol booster.

5. Acknowledgement

This research is partly funded by the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium) and by the Belgian Science Policy through the IAP V/11 contract.

6. References

[1] IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999) IEEE Standards for Information Technology -- Telecommunications and Information Exchange between Systems -- Local and Metropolitan Area Network -- Specific Requirements -- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

[2] IEEE 802.11b-1999 Supplement to 802.11-1999, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band

[3] IEEE 802.11 WG, Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/Draft 4.2, February 2003.

[4] Fischer M. Hybrid Coordination Function (HCF) frame exchange and NAV details, *IEEE 802.11e working document* 802.11-01/109r2, 2001.

[5] Zorzi M, Rao RR, and Milstein LB. ARQ error control for fading mobile radio channels, *IEEE Transactions on Vehicular Technology*, 1997; 46(2):445-455.

[6] Feldmeier D. C., McAuley A. J., Smith M., Bakin W. S. Marcus and Raleigh T. M. Protocol Boosters, *IEEE Journal on Selected Areas in Communications,* 1999, 16(3): 437-444.9

[7] Hoebeke J., Van Leeuwen T., Peeters I., Cooreman K., Moerman I., Dhoedt B., Demeester P. Development of a TCP protocol booster over a wireless link, *SCVT 2002 Louvain-La-Neuve*, 2002, p27-34

[8] Click Router Project http://www.pdos.lcs.mit.edu/click/

[9] Gilbert E. N. Capacity of a Burst Noise Channel, BSTJ September 1960.

New Wine into New Skins: Exploring the Educational Affordances of Ubiquitous Computing....109
Tony Hall, Liam Bannon, Eamonn Murphy, UL

An Evaluation of Mouth to Ear Delays caused by VOIP Endpoints....119
Maria Varley, Hugh Melvin, NUIG

Technical Session D – Wireless Sensor Networks and 802.11....129

Predictable and Controllable Wireless Sensor Networks....131
Utz Roedig, Cormac Sreenan, UCC

Enhancing QoS in IEEE802.11e WLANs Using Cell Breathing....139
Olivia Brickley, Susan Rea, Dirk Pesch, CIT

An UDP Protocol Booster for Mukltimedia Communications over 802.11 Wireless LAN....151
Abram Schoutteet, Jeroen Hoebeke, Tom VAN Leeuwen, Ingrid Moerman, Bart Dhoedt, Piet Demeester, Ghent University, Belgium, Nico Janssens, Pierre Verbaeten, K.U. Leuven

Technical Session E – Wireless Network QoS....163

System Initiated Context Modification to improve Network Resource Usage....165
Utz Roedig, Cormac Sreenan, UCC

Network Selection Strategy in Heterogeneous Wireless Networks....175
Olga Ormond ,John Murphy, UCD, Gabriel Muntean, DCU

Industry Track....185

Perception: The Real Inhibitor to ASP Adoption?....187
Sabrina McNeely, Kenneth Kirrane, John Healy, Sean Duignan, GMIT

Overview of Wi-Max IEEE 802.16....197
Alan Barry, George Healy, Cian Daly, Joseph Johnson and Ronan J. Skehill, UL

A 3G Mobile Services EcoSystem : Catalyst....205
Tom Pfeifer, Barry Downes, Peter Elger, TSSG, WIT

Information Technology & Telecommunications

IT & T

IT&T
Annual Conference

"Information Infrastructures – Systems, Architecture and Content"

26th & 27th October
National Maritime College,
Cork Institute of Technology,
Ringaskiddy, Co. Cork Ireland

Proceedings

ISSN 1649 – 1246

4.00-4.10	*Coffee Break*		
4.10-5.30	**Technical Session C Ubiquitous Computing** Chair - Dr Jimmy McGibney, TSSG, WIT *Applying Iterated Local Search to reduce the costs of backhaul in telecommunications network design, Joe Bater, UCC* *A Software Toolkit Library to support rapid prototyping of novel interaction techniques, Paul Gallagher, UL* *New Wine into new skins: exploring the educational affordances of ubiquitous computing, Tony Hall, UL* *An Evaluation of Mouth to Ear Delays caused by VOIP Endpoints, Maria Varley, NUIG*	**EU Management Workshop Contd.**	**Industry Track session II contd** *A 3G Mobile Services EcoSystem: Catalyst Tom Pfeiffer, WIT*
5.30 - 6.30	*Posters demonstrations and Wine Reception- Dr Irene Sheridan, CIT*		
8.30	*Conference Dinner – Carrigaline Court Hotel. Pre-dinner drinks in the Kingfisher room @8pm*		

THURSDAY 27TH OCTOBER

9.15 - 10.15 *Technical Session D*
Wireless Sensor Networks and 802.11
Chair – Dr Dirk Pesch, CIT

Predictable and Controllable Wireless Sensor Networks,
Utz Roedig, UCC

Enhancing QoS in IEEE802.11e WLANs Using Cell Breathing,
Olivia Brickley, CIT

An UDP Protocol Booster for Multimedia Communications over 802.11 Wireless LAN,
Abram Schoutteet, Ghent University, Belgium

10.15 - 11.00 *Keynote Address*
Tim Stone, Cisco Systems Europe
Chair – Jonathan Evans, ComReg

IP Communications- Changing the way Businesses Communicate
Convergence of Voice, Video and Data is now mainstream for European Businesses. The most important driver over the past 5 years has been around cost savings and return on investment focussed on the IT department and network infrastructure. The market is now moving into a new phase, business leaders are increasingly trying to understand the employee impact of deploying new converged communications applications. This presentation highlights the development of the market for convergence, the organisational impact, and looks at key technologies that will impact the way businesses communicate in the future.

11.00-11.10 *Coffee Break*

11.10 - 12.30 *Commercialisation Session*
Chair - Gearóid Mooney, Enterprise Ireland
Billy Harkin, CEO Science Ventures
Willie Donnelly, TSSG, WIT

Commercialisation of research is often seen as a response to increased research investment. What is meant by commercialisation and how best can the third level sector turn ideas into profit? In fact is it possible for this to happen at all as academia and business may be mutually contradictory terms! Share the experiences of those responsible for this onerous undertaking.

12.30 - 1.10 *Technical Session E*
Wireless Network QoS
Chair - Dr John Murphy, UCD

System Initiated Context Modification to improve network resouce usage,
Utz Roedig, UCC

Network Selection Strategy in Heteregeneous Wireless Networks,
Olga Ormond, UCD

1.10 - 1.40 *Poster Presentations*
Chair - John Fagan, Enterprise Ireland

1.40 - 2.45 *Lunch and Prize giving*