Packet Scheduling in Wireless Network IEEE802.11 Using Linux

Johan Cimen

Master thesis work for degree in
Master of Science Programme in Computing Science and Engineering
Department of Computing Science, Umeå University,
Umeå, Sweden

7th April 2004

Supervisor: Thomas Nilsson
Examiner: Per Lindström

Abstract

In the downlink of a wireless network standard IEEE802.11b, there is fair treatment of flows, which makes that all flows receives an equal amount of the bandwidth capacity. This is a limitation in the standard, because it cannot control the available bandwidth for prioritising flows dependent on the required service. Especially in the performance domain when congestion, interference, location dependent errors and drop of packets leads to reduced service quality. This master thesis work is for insight in improvement of network service in a wireless LAN and how to achieve quality of service. The purpose was to build a packet scheduler in a Linux environment that hierarchical scales the downlink capacity and schedules traffic flows based on signal strength value that stations perceive. Test results show that the Hierarchal Token Bucket scheduling algorithm for packet flows, created for wired network, performs well in the wireless network. The Quality of Service was improved when preferential service was given for flows, using differential service for resource allocation and signal quality values to schedule aggregate flows to stations.

Contents

List of Figures

List of Tables

Acknowledgements

Thanks to Lars Lyxel at Testing Vital for his support in hardware problems and loan of laptops with wireless equipment, which made it possible to design and test packet scheduling.

Thanks to Heratch Ayvazian for his help in proofreading this report.

Thanks also to my supervisor Thomas Nilsson for his support.

1 Introduction

In a Wireless Local Area Network (WLAN) using the IEEE802.11 standard[1], the topology is called infrastructure-based if it consists of a group of stations associated with an Access Point (AP). AP is the central communicator, which bridges packets among stations and fairly distributes the downlink traffic capacity to stations. Transmissions from AP to stations is controlled only in the meaning that every downlink flow has the same opportunity to send its packet and also receive an equally share of the downlink capacity [19]. There is a demand to control the downlink for prioritising flows from particular stations or for flow requiring special service. Reason for this is that; equally sharing the downlink capacity among existing flows in the IEEE802.11b standard, does not meet requirements from applications for more resources when needed. Real-time applications and streamed flows for multimedia traffic are examples of flows that are treated the same as all other flows and therefore receives the same amount of bandwidth. These and other types of applications that have more than common requirements on resources cannot work properly in such circumstances. Other limitations wireless communication experiences are: location dependent errors and multi path propagation [29] [15]. These limitations affect the outgoing service and complicate construction of a Quality of service (QoS) for a wireless network.

With a centralised AP, that coordinates the downlink traffic and at same time being one among all stations that contends for access to the medium, makes resource sharing on the downlink from AP complicated. Scheduling algorithms for aggregate traffic that are created for wired network cannot always be used in a wireless network with these aspects of wireless communication limits and medium access contention. Suggestion to obtain high service in a wireless network is: an ability to control flows using classification and after that scheduling dependent on classes. This refers to the decision process to determine which flows shall be served with more resources and how the downlink capacity shall be shaped into classes to guarantee a quality of service.

This report will explain how to control the downlink by scheduling traffic and how to give preferential service to classified flows. It will also explain how to implement QoS in differentiated service adapted for a wireless network standard IEEE802.11b in Linux. Methods used in wired network to obtain QoS with: filtering, policing, queuing and scheduling will be used in wireless network. Filtering rules will be created to classify packet flows and quantities will be defined to hierarchical divide the downlink capacity. Rules for classification of packet flows will be based on: signal quality stations receive from AP and packet type settings for

[1] http://grouper.ieee.org/groups/802/11/index.html (Visited 2004-02-23)

QoS. To classify flows based on these premises, every station in the WLAN has to measure the signal quality and report it to AP within packets on the uplink. A channel monitor will be used at AP to store signal quality values received on the uplink and to provision the scheduler with these values on the downlink. Results will show that downlink can be controlled.

2 Packet switching

In a wireless as well as in a wired network, stations are communicating using a communication protocol. The most known communication protocol in Internet is: Internet Protocol (IP). This protocol has specification of format and other control information in the packet header (Figure 1), which is used by switches and routers in a network to direct packets and also by stations to set values into.

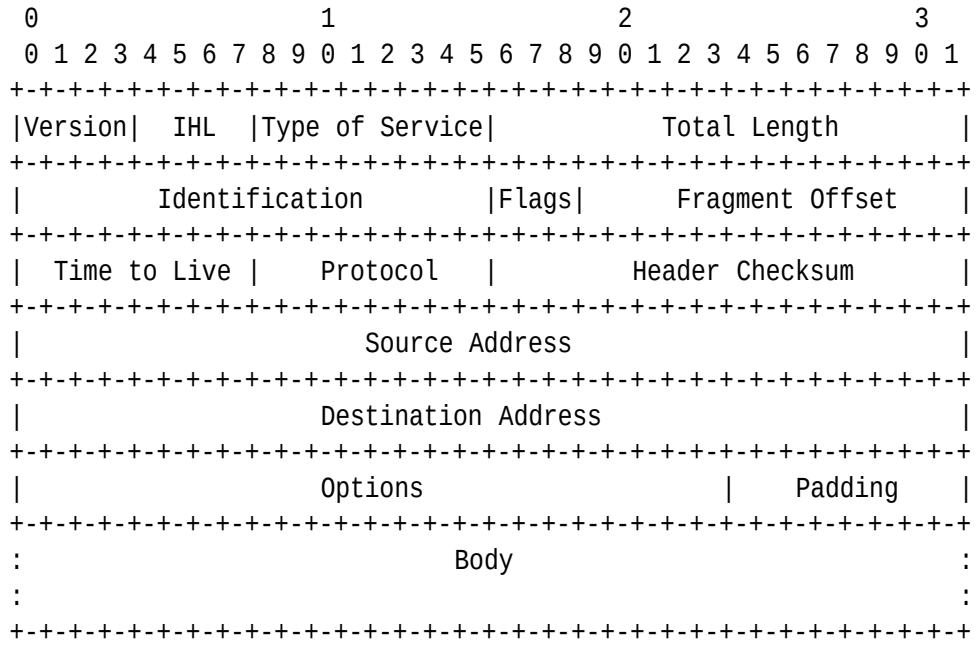

Figure 1: Internet Protocol header

The User Datagram Packet (UDP) protocol is a transport protocol used to extend the IP-protocols delivery service. UDP is as IP a datagram service, which means that there is a sender and a receiver of the packet in a communication. UDP does not require connection handling between sender and receiver, which yields that packet transmission can start asynchronous from a sender in the network to an existing destination. This protocol has no reliability, since it has no connection handling and this gives no guarantee that UDP packets arrive to their destinations, but the simplicity of it makes it useful in tests and measuring of network capacity.

A network application that sends an IP-packet, using the UDP datagram, includes the messages into an UDP packet in the transport layer and then sends it to the network layer (Figure 2). At network layer it is wrapped into an IP packet and after that linked to the physical layer where the network card transmits it via the outgoing queue. The same procedure happens in the other direction when packets are received at the destination. This is the standard procedure of transmitting and receiving UDP packets.

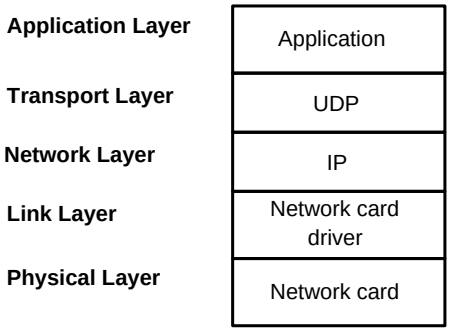

Figure 2: IP network stack.

3 Wireless network

Communication in a wireless network built with the IEEE802.11b standard and using the IP protocol happens in same way as in a wired network. Difference is that stations in a wireless network are sending packets through the air medium with radio signals to an AP. AP is a bridging node that directs packets in a Basic Service Set (BSS) topology where there is one AP connected to a wired network (Figure 3). All communication from stations goes through AP into the wired network as well as between stations.

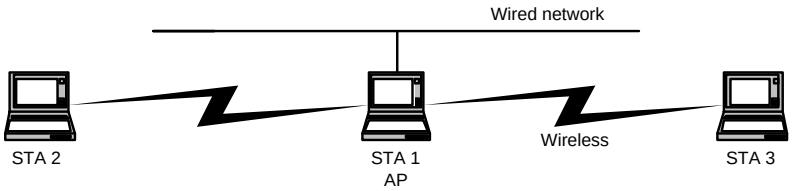

Figure 3: BSS network.

In the IEEE802.11b standard, there are two sub layers in the Data Link Layer of the protocol stack (Figure 4). The Media Access Control (MAC) protocol controls access to the wireless medium. It preserves synchronisation between stations and AP by transmitting beacon frames to stations and it also manages power consumption during transmission [9]. The Logical Link Control (LLC) handles communication errors and retransmissions in the channel.

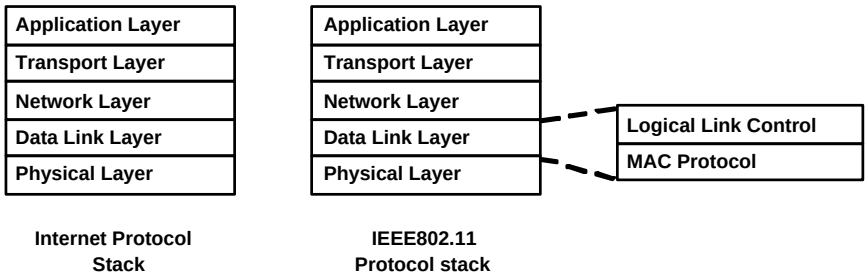

Figure 4: Comparison of IP- and IEEE802.11 network stack.

The AP is the central communicator in a BSS topology and all the communication from stations goes through the AP. Communication on the uplink, when stations transmit data to AP is half duplex, since stations cannot listen and transmit data in the channel at the same time. The Distributed Coordination Function

(DCF) mechanism in the MAC layer, controls access to the medium on the uplink (Figure 5)[11]. The DCF function is a random access protocol based on the Carrier Sense Multiple Access (CSMA) with algorithm for Collision Avoidance (CA) [19]. This method with CA shall prevent collision of packets in transmissions when all stations are contending for the medium.

A station in the wireless network that wants to transmit data, performs a carrier sense of the medium first. If the medium is idle, it waits a time period of a DCF Inter Frame Spacing (DIFS), which is a time period a station has to wait before sensing the medium again. If the medium is idle again after this DIFS period, data will be transmitted immediately. The receiver of data waits a Short Inter Frame Spacing (SIFS), which is a shorter time period than DIFS and is used with acknowledgements in the communication. After the SIFS, an Acknowledgement ACK is replied back to the sending station. During the time when a station has access to the medium, all other stations are deferring access.
Otherwise, if the sender had sensed the medium busy it had to waits a DIFS and after that entered the contention phase. In this contention phase, each station has to generate a random backoff time and wait for this time period. The contention period is counted in slots and after each elapsed slot the backoff value is counted down, but only when the medium is idle. If the medium is idle after the backoff time period, a station has to wait a DIFS as described above and transmit after that if it is still idle.

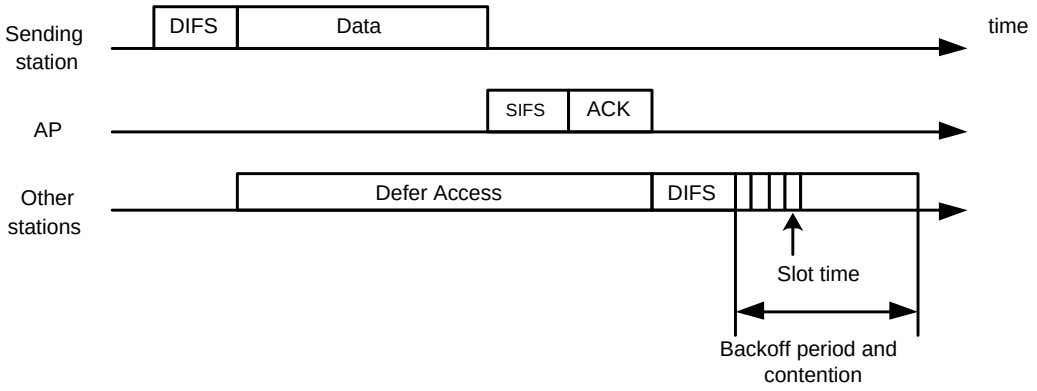

Figure 5: Medium access control procedure using DCF.

4 Quality of service

Providing high quality of service in a wired as well as in a wireless network should be the very next important task after obtaining functionality. The definition of service here is in the meaning of a pre-defined treatment of data during their existence in the network and to guarantee requirements of service where available resources in the network are shared among traffic flows. In a wireless network where AP is bridging packets, there is fair processing of traffic flows on the downlink. This is a service in the definition of per flow, where each flow has an equal opportunity to be transmitted and receives an equal amount of the transmit capacity. The quality of this service is a limitation for applications that requires more resources than regular traffic. Besides the given service in form of resource, there are environmental aspects in a wireless network that will affect the quality of a given service, which will be explained in Section 5.1.

One of these factors that limit the QoS given to stations in a wireless network is the varying quality of radio signal between stations and AP within a BSS. Multimedia traffic is one type of flow that is mission critical and sensitive for variations of signal quality. Other types of traffic with properties that needs to be distinguished in a network are: delay sensitive traffic, high throughput requiring traffic and reliability requiring traffic. Selections among these types of traffic flows are not possible using the IP protocol, because all packets are treated equally. Distinguishing of traffic flows to give special treatment can be done at either side of the communication path. With a traffic controlling mechanism at sending or receiving station, flows can be distinguished to achieve preferential service and to be prioritised.

A traffic controlling mechanism that filters flows is a scheduler with a scheduling policy to manage resource sharing dependent on traffic type and network specific requirements. A scheduler can give QoS in a wireless network, where flows can be served in an ordered manner and traffic can be handled in flow-specific time constraints to prevent latency and loss. A scheduler can also get rid of the present situations of an IEEE802.11b with fair processing of traffic flows and especially when a particular flow causes head-of-line (HOL) blocking. A HOL situation occurs when there are aggregate flows that are sharing one outgoing packet queue and this becomes blocked, caused by one or more reasons that are going to be explained in Section: 5.1. HOL blocking results in that all downlink flows suffer additional delay in the queue.

Five principles that build QoS are presented by Andrew T. Campbell [8]:

Integration principle: This principle states that the service shall be configurable in all IP architectural layers to obtain quality of service in an end-to-end transfer.

Separation principle: Classification of packets to give preferential and required service.

Transparency principle: The application has to be free from underlying components that are used to obtain QoS. A user application is declaring what type of service is required, not how this shall be achieved.

Asynchronous resource management principle: Functionality of QoS is divided into architectural components consisting of controlling and management modules, which sometime collaborate asynchronous.

Performance principle: Rules and recommendations of traffic creates order and structure to the communication protocols for functionality and high performance.

4.1 QoS mechanisms

4.1.1 Integrated Service

A network, using IP-protocol is defined as best-effort service, which means that the protocol makes its best to transfer packets from sender to receiver and there is no guarantee that packets in the network reaches their destinations [20]. The best effort service in the IP-network does not suit all traffic types in the network. There are applications requiring resource reservation and prioritising of particular type of flows to give QoS. Prioritising traffic flow and resource reservation for an application at a particular stations can be achieved using the Resource Reservation Setup Protocol (RSVP) [6]. The RSVP is a signalling protocol for resource reservation in the Internet architecture and it is a framework developed by Internet Engineering Task Force (IETF) [26]. This protocol provides QoS to certain stations where the requested resource reservation from a station is in one direction and it uses traffic controlling mechanisms to control the traffic. The packet classifying mechanism classifies packets into QoS classes, policy control mechanisms checks if the requesting station has permission to make a reservation, the admission control mechanism has control of the available resources and the packet scheduler mechanism provides the agreed resource to stations.

In a simplex communication, where there is one data transfer direction, a station starts with asking the control mechanism at the resource sharing node for resource reservation [33]. In the request, there is a traffic specification message-signalling packet, which specifies characteristics of sender's data and a resource specification message-signalling packet, which is for resource reservation. If the reservation is accepted, there is a one-way resource reservation agreement and the station starts to transmit packets to the router. Arriving packets to router are classified into classes and treated dependent on packet type and the agreement in RSVP. Classified packets are then scheduled through the scheduler and transferred further to next router or to their destinations. In an end-to-end situation with QoS in both ends, the RSVP negotiation with signalling must be done in both directions.

The Integrated Service architecture (Interserv) provides defined service to traffic flows with certain QoS commitments using RSPV signalling. Interserv provides two services: guaranteed and controlled-load service.
The guaranteed service guarantees that packets arrive to their destinations and is intended for real-time applications. It uses both traffic specification message and resource reservation message signalling packets to police requirements. The guaranteed service is provided using a token bucket method to reshape the traffic flow into the traffic specification. A token bucket method regulates the traffic flow using a buffer (bucket) of tokens for maximum rate. Tokens are generated into the bucket with a rate and data of size S can only be sent if there are S tokens in the bucket (Appendix D).
The Controlled load service provides no guarantees there is only a high percentage of delivery of packets [30]. There is only a traffic specification-signalling message used for flow specification. This service type is intended for traffic flows that can tolerate a certain amount of loss and delay.

4.1.2 Differentiated Service

Differentiated Service (Diffserv) provides QoS in a network to preferential identified classes of traffic flows. In difference to Interserv, where there is signalling for per session, the Diffserv setup is static at network nodes, which builds domains and are therefore for long term. Traffic flows entering a boundary node are classified into behaviour aggregate classes and these classes designate treatment of packets in the flow dependent on service level agreement.

There are two essential components in Diffserv for an end-to-end implementation: Marking and Per Hop Behaviour.

- **Marking**
  Instead of signalling message in the network, which takes network capacity, the Type of Service (TOS) octet in the IP-header is marked to declare what type of treatment the classified flow shall have (Figure 1). The TOS-field is eight bits long and the two right most bits; bit 7 and 8 are currently unused, so there are only six bits used to describe packet type and treatment (Figure 6) [24]. This six-bit field called; Differentiated Service Code Point (DSCP) is divided into two parts: the three left most bits; bit 0 to 2 are for IP precedence and the three right most bits describes the Diffserv class selector code points, which means the treatment of the packet.

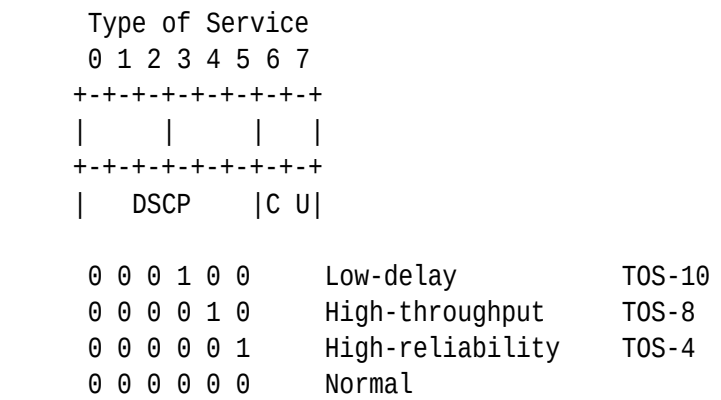

Figure 6: DSCP-field with EF-codes.

- **Per Hop Behaviour**
  This component refers to the service that will be given for packets dependent on the DSCP code. Packets arriving to a boundary node of a Diffserv domain are filtered into classes based on the DSCP code. This code tells what type of service and treatment the packet shall have in its Per Hop Behaviour (PHB). The PHB is information to the Diffserv node about queuing and scheduling treatment of a particular packet. After classification, service will be given for that class dependent on configurations for resource sharing and service level agreements [4]. The PHB preserve two types of services: Expedited Forwarding and Assured Forwarding.

  - **Expedited Forwarding (EF):**
    This service assures that packets will be forwarded from the node (node can be a bridge or router) with low delay, low jitter and gives

high guarantee that packets arrive to their destinations (low loss probability) [12]. Figure 6 shows four examples of DSCP codes with marked EF bits. The minimize-delay value is for applications such as: FTP, Telnet and SSH, that wants to minimize delays on packets through network nodes. Application like FTP-data, requiring fast service for its packets through the network, marks them for high-throughput. Packets for reliable links through the network are marked with high-reliability. Default is that non-marked packets have DSCP code '000 000'. Packets from stations that are sending regular packets without DSCP-value are classified as normal traffic.

– **Assured Forwarding (AF):**
This service does also offer special service better than best effort, but the service level is lower than that in EF. It has four defined classes for behaviour aggregate traffic. Each class has a buffer space and bandwidth specified in the network interface. For each class there are three drop-precedence values, which will be used to drop packets when there is congestion [17] (See DSCP codes for AF at Appendix C). With four classes and three-drop levels, there are 12 combinations for a packet in a AF service. When there is congestion in a network packets with high drop precedence will be dropped first before medium and finally low drop precedence.

The simplicity of Diffserv without signalling protocol for resource reservation and classification of flows makes it easy to implement and it has scalability, which Interserv does not. Diffserv has also transparency; application has no relation to the resource sharing that takes place on the lower network layers.

4.2 QoS performance

Performance of service in QoS is a non-functional property, which represents rules and structure (Section 4). A degree of QoS in a system is a service performance agreement between involved parts and operating parameters that are defined to measure QoS are [18][10]:

- **Bandwidth**
  Requirement of this parameter from applications is hard to fulfil without a scheduler. In a scenario where there is fair bandwidth sharing, the quantum of bandwidth given to a flow will be varying dependent on number of flows sharing the bandwidth. This makes it hard to give a quality of service. To ensure a guarantee of minimum bandwidth reservation for flows and to ensure QoS in a network, there is necessary with a bandwidth managing scheduler at the resource sharing network node.

- **Latency**
  Latency value is the time period for a packet on its journey in the network from sending moment until it reaches the destination. Arrival of packet has to be timely and not varying too much for delay sensitive applications like IP-phone or streamed multimedia data. Increasing latency values can reduce the service quality value. A factor that affects the latency value is buffering at network nodes like: routers and bridges, but also at stations outgoing and incoming queues. Buffers are used in a network when nodes cannot handle packets immediately [7].

- **Jitter**
  Jitter is related to latency and is the variation of the time period between adjacent arriving packets [2]. Buffering at receiving station can control jitter, which is done in most multimedia applications. This buffering is different from previous mentioned latency buffer. Jitter buffering it is for application performance, used for backlogged flow at sender and receive buffer at receiver for appropriate usage of data without arrival variations. Measurement of jitter values is complicated, since sending and receiving stations must have synchronised clocks.

- **Losses**
  Packet loss rate is a parameter that describes the reliability for a communication link. Packets can be lost of several reasons in a wireless network. One reason is congestion at an end of a communication path causing HOL, which creates delay of packets and drop if there are time constraint packets. Other reasons are electromagnetic interference, noise or location based errors in a communication link, which also will cause drop of packets if there are bit errors in received data. More about packet dropping factors will be explained in Section 5.1.

[2] Definition of this term is used in different ways. In [14], it is defined as variations of packet delay.

5 Packet scheduling in wireless network

The IEEE802.11b standard for wireless networks uses the Industrial Scientific and Medical (ISM) frequency band 2,40-2,48 GHZ for communication. This frequency band is divided into 13 channels in Europe and there are national directions for channel frequencies [19]. IP-packets sent from a station or AP is first converted from bit streams to radio signal, which is modulated to a frequency for a channel and after that transmitted into the air medium. All network nodes within the BSS network are using the same channel to communicate and all stations within the LAN have equal opportunity to access to the medium using DCF protocol, including AP. AP, which is the bridging node has no privileges to transmit more on its downlink when sharing resources, but all flows from AP are treated equally and there is no contention among these. There is only one outgoing queue, which is a First In First Out (FIFO), where aggregate flows are queued in an arrival manner. With N flows on the downlink, each flow receive 1/N of the available bandwidth capacity fairly.

5.1 Wireless channel

In a comparison to a wired network where communication is within the wire, the wireless channel suffers from environmental impacts and appearance of packet errors that limits the service level. A main factor affecting the channel condition is the distance between station and AP. Transmitted signal has propagation dependent on the antennas purpose. For a common antenna, it is circular in all directions and the signal strength will fade with the distance, d from the sender (Equation: 1) [27]. Station with a distance d at position p, will receive a higher signal strength than a station with distance d' at position p', where d < d'.

$$Received\ signal\ strength = \frac{1}{d^2} \tag{1}$$

Stations that are sharing same channel will perceive different signal strength dependent on factors that impacts on the signal quality and limits the channels utilization capacity. Path loss and fading are impacts, which can appear differently for stations dependent on position and distance to AP [15]. Schiller explains in [29] that an emitted radio signal does not always take the straight line to its destination. Reason for this is obstacles between sender and receiver, which cause multiple paths called multi path propagation. Multiple small signals created from one single transmitted signal, yields varying arriving time and strength at receiver. Environmental factors like: multi path propagation, electromagnetic interference

and noise cause bit errors in the modulated radio signal, which leads to discarding of erroneous packets at receiving station. Communication capacity can be measured in wireless network using the Bit Error Rate (BER), which gives a percentage from: amount of bits with error among all received bits. A drawback of this capacity value is that higher transmission rate gives higher BER when the amount of bits in a period increases in ratio to the noise [31].

Non-environmental factors like location dependent error, can according to Harilaos et al. [28] (2002), among stations sharing the same channel have different errors dependent on their location. An error dependent on a location has a fault at either one or both ends, which will give an unclean channel and can cause burst error [2]. Traffic burst is the amount of packets that can exists in the channel during a time period. To many packets will cause error and packet loss when network nodes cannot handle the amount of packets. In UDP traffic where there is no connection handling and the only control is packet checksum at packet destination, the burst error will affect the application.

Representation for an accurate signal quality value in the BSS that stations receive has to be measured at stations. The location based measured value will include both environmental and a location dependent error at that position each station has. The received signal strength value including interference and noise impact from the environment can be measured using the Signal to Noise Ratio (SNR). This value gives the signal quality value in ratio to measured signal strength and noise level in the radio signal. This will be a better choice than using BER, since BER gives the rate of bit errors in the link and does not representing the radio signal quality.

5.1.1 Signal strength

The strength of a transmitted signal is attenuating with an increasing distance to its destination. Received signal strength at a station will be lower than transmitted and it will also include noise [5]. Explanation in Section 5.1 described that; besides the distance from station to AP, environmental factors impacts on a signals propagations and affects the signal quality. The transmitted signal power value is given in the unit Watt (W), but the radio signal is not spreading linear and therefore there it is calculated in logarithmic comparison value. Comparison is between: power value at transmission from AP (P1) and the received signal strength at station (P2) [5]. This comparison will give a factor of loss of signal in the unit of decibel (dB). A wireless network cards with reference level of 1mW

source power will give the dBm unit (Equation: 2).

$$Signal\ strength\ (dBm) = 10\log 10(\frac{P2}{P1}) \Rightarrow 10\log 10(\frac{P2}{1mW}) \quad (2)$$

There will be an attenuation of the signal strength during the propagation and the receiving station will measure a lower signal value P2 than transmitted. The dBm result, characterising the signal strength will therefore be negative (Equation: 2). A high result will represent a high signal quality, which is good radio signal value with low environmental impact.

Stations will measure radio signal strength (P2) from AP in a situation having the distance 'd' to AP and when receiving data traffic. The noise level (P1) in the environment will be measured when there is radio silence from AP. The SNR-value (S/N) is then estimated in dB (Equation: 3) using these two values.

$$SNR\ (dB) = 10\log 10(\frac{P2}{P1}) \quad (3)$$

Theoretical channel capacity can be achieved using the Shannon Capacity Formula in Equation: 4, where the B constant is the bandwidth of the channel in Hertz[31]:

$$Capacity\ (Mbps) = B\log 2(1 + SNR) \quad (4)$$

5.2 Monitoring wireless channel

To provide high QoS in a wireless network where the channel quality can change rapidly, caused by radio signal propagation and environmental aspects (Section 5) there is a need for a channel monitoring system that can verify network quality. The channel can be monitored to be prepared for channel variations and prevent complications that can be caused in the communication. Channel monitoring can be achieved by active tests or via passive monitoring. Active testing is when the resource manager tests the channel in intervals for updating the condition status. Passive monitoring is when stations report the channel condition to a centralised monitor at the resource manager. With an already centralised AP, which controls the medium access in downlink and where the downlink resource will be managed with a packet-scheduler, it would be a good choice to use passive monitoring of the channel at the AP [23]. To provide the monitor with accurate channel quality each station within the BBS-service system has to measure and calculate the quality of received radio signal and after that report it to the AP. Problem that arises after the channel quality estimation is how this information shall be delivered

to the AP. The only communication between stations and AP in a BSS is via sending and receiving data packets. The IP protocol used in the wireless network is designed for wired networks and there is no field for the wireless channel quality value within the packet-header. One solution is to use a piggyback value included in outgoing packets to AP. The drawback of this solution is that it will be an overhead behaviour with another processing of including value, calculating the packets checksum at sender and reversal processing at receiver. Instead of using a piggyback value, an already defined packet header field can be modified for this purpose. The TTL-field is not used in a BBS topology and especially not in an end-to-end communication between station and AP (Table 7). The Time-To-Live (TTL) field in the IP header indicates the time a packet can stay in the system. For each router or bridge the packet is passing, the TTL value will be decreased by one and a packet having zero value and which has not arrived to its destination will be dropped [13]. The usage of the TTL-field for delivering the link quality value to the AP is limited to the BSS network. Packets outgoing to the wired network, forwarded by the AP have to be rewritten to a default value for the TTL-value. This can be done in AP: s outgoing packets, which will be described later in the networking function Netfilter. Measurement of the current link quality will be

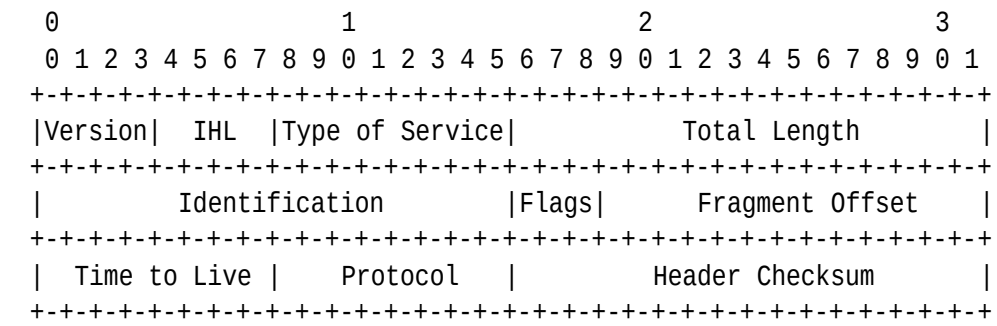

Figure 7: TTL field in IPv4 header

at stations because, AP cannot obtain the channel quality value with the location dependent error the station has with its perspective. The monitor at the AP will log the quality value of radio signal received within packets and representing stations.

This representing value can then be used in downlink scheduling of flows back to stations. Stations that had good signal quality can receive more resource than a station having bad quality value. Using the quality value for scheduling does not tell anything about required service for packet flow on the downlink and using only QoS given DSCP codes gives service but there is no knowledge about the link quality to stations. A combination of the signal quality value that stations sends to AP and the DSCP value in the TOS-field for PHB, should be a good combination for incrementing the QoS to stations.

5.2.1 Link quality estimation

The link quality measurement and estimation that are going to be used in this project is the same method that was used in a spin-off project [3] (2003). The radio signal strength measured by stations will first be quantised to a category between 15 and zero. Categorised value will then be converted to binary value before inclusion into the TTL-field of the IP packet-header. The TTL-field of a packet-header consists of eight bits and the largest value that can be inserted into this field has the decimal value 256.

Signal	Binary	TTL-Binary	TTL-Decimal
0	0	11000001	193
1	1	11000011	195
2	10	11000101	197
3	11	11000111	199
4	100	11001001	201
5	101	11001011	203
6	110	11001101	205
7	111	11001111	207
8	1000	11010001	209
9	1001	11010011	211
10	1010	11010101	213
11	1011	11010111	215
12	1100	11011001	217
13	1101	11011011	219
14	1110	11011101	221
15	1111	11011111	223

Table 1: Signal quality value conversion.

The conversion Table1 shows the quantified signal quality value and corresponding binary value. Column two shows the binary signal value included in the TTL-field bits two to five. The Most Significant Bit (MSB) and the Least Significant Bit (LSB) of the TTL-field set will indicate that this is a signal quality value included in this field. Values in the fourth column are the corresponding decimal value to TTL-binary values, which is shown in network logs when logging.

All these operations of: link quality value measurement, signal value calculation, conversion to binary and insertion into a packets TTL-field is done at the wireless network card driver at a station.

5.3 Controlling traffic

With a signal quality value and a monitor that can store values it is necessary to organise packet flows at AP for link sharing and optimal usage of resources. Packet control can be done at both ingress to AP and egress. Controlling is not only shaping the receiving and sending rate in favour for prioritised flows; it is also to schedule and achieve QoS. Linux kernel-2.4.20 has a traffic control system called Traffic Control (TC) included in the network layer (Figure 8). This system can be used for support of various provisioning methods and managing the bandwidth in a network. Packet queues can be applied into the kernel for classification, prioritising, scheduling and shaping the outbound traffic [3]. Main elements of TC are queues, classification and scheduling.

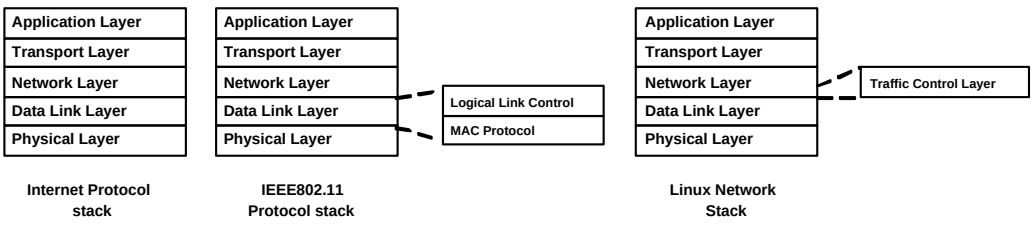

Figure 8: IP, IEEE802.11b and Linux network stack.

Traffic Control New Generation (TCNG) is a revision of TC and is developed to be more extensible by supporting different implementation languages like C, Perl, Java and XML. TCNG is like TC; providing network management and mon-

[3] LARTC, Linux Advanced Routing & Traffic Control HOWTO, http://lartc.org/ (Visited 2003-11-27)

itoring components, which mean that it is possible to build a router or bridge [4]. This new revision is for easing the construction of traffic controlling systems by offering several programming language and a better usability of TC. Instead of TC rules, there is now understandable syntax: "action operator expression" and programming structure instead of line rules. With those mentioned programming languages; it is possible to compile TCNG code with tcc compiler to TC executable shell script. Executing script will load TC commands into the Linux kernel. It is now possible to build robust traffic controlling architectures controlled by a compiler.

TCNG has been used in this project to build a packet-scheduling platform for prioritising packet flows dependent on the signal quality value stations perceives. The scheduler will also provide QoS when flows are specified with QoS parameters.

5.3.1 Queuing disciplines

Each network device on AP can be assigned one queue discipline (qdisc) to control outgoing packets. A qdisc is called classful queue if it has classes of queues within it, where packets are queued. A classless queue is just a queue, nothing more [1]. In the same manner, a qdisc within a qdisc can have a queue and is called classful. The characteristics of the classful queue built with queues determine the way packets are going to be sent from AP and are therefore closely tight to the scheduling algorithms behaviour. Classful queues are useful when there are consideration for different type of traffic in the network. Each traffic type can be queued into a classless queue within the classful queue . Classless queues defined in TC are:

- **First In First Out (FIFO)**
  The most known classless queues is First In First Out (FIFO). It has a simple function in: first arriving packet is dequeued first and those packets arrived after are dequeued in an arrival order.

- **Token Bucket Filter (TBF)**
  This queue has a controlling method using tokens and a leaky bucket. Tokens are dropping down into a bucket in a controlled rate and the leaking hole of the bucket is also controlled. For each token leaking from the bucket: one Byte, packet or another predefined size of data is dequeued. This queuing method with control can be used to shape packet flows.

[4] Linux Traffic Control Next Generation, http://tcng.sourceforge.net (Visited 2004-02-23)

- **Stochastic Fair Queue (SFQ)**
  SFQ has a fairly distribution of service in the queue to all flows. Packets arriving to the queue are sorted by flows into FIFO queues and these queues are then dequeued in a Round Robin (RR) fashion.

5.3.2 Classification

This is the definition for packet queues, which consist of packets that have been classified with a filter. Classes can either be classful or classless and are identified with a 'class' name. Classes are used by a queue discipline to manipulate packets to perform: shaping, scheduling and policing.

- **Filters**
  Filters are used to classify packets into classes or qdiscs dependent on rules and the desired design. These rules can for an example be based on QoS parameters or fields in the packet header:
  - Link condition in the channel
  - Packet type
  - Source/destination port
  - Source/destination address
  - Low delay packets, having TOS-value 10
  - High throughput packets, having TOS-value 8
  - High reliability packets, having TOS-value 4

  Filtering rules can be built like a domain relation question in a database system [16]. Starting with a brief filtering like: all packets having TTL-value above 193, and continuing to narrow filtering: having TOS-value 10 AND destination to port 7001, will give specific selection of packets. These specification-based rules are grouped to classifier groups:
  - **u32 filter**
    This filter group uses the octets in the IP-header to distinguish packets. Filtering elements are for instance: protocol, source or destination address, source or destination port, MAC-address, TTL-value or TOS-value.
  - **route**
    When the boundary node of the Diffserv is a router or bridge and arriving packets is to be forwarded to a domain node or to another boundary

node these packets are filtered. Filtering those packets at a boundary node gives them another type of service and resource.

– **Differentiated Service MARK (DSMARK)**
This filter rule belongs to Diffserv implementations and is used to set a bit in the TOS-field when packets are entering a Diffserv domain. DSMARK is also a qdisc, which only mark and classify packets based on DSCP-codes.

5.3.3 Ingress

In an end-to-end communication, when packets are arriving to AP, theses are passed to the TC-layer and ingress. Ingress is a collection of packet handling operations defined in TCNG for incoming packets. Packets enter ingress in arrival order and are filtered to classes. Packets in the class are of a specified type and can now be policed to a defined rate with SLB. Packets not fitting a filter can either be classified into a best-effort class or be dropped.

After filtering, each class consists of a specific type of packets that can be: forwarded to a node, dropped, reclassified or delayed. Policing is a process of shaping flows by a certain criteria. Packets passing the shaper are for a certain required activity and if not; packets are discarded [25]. Single Leaky Bucket (SLB) is one of the policing mechanisms in TC to reshape packet flows. The SLB is most like the TBF that was explained for classless queues. Packets are passing the rate controlling mechanism if there are tokens in a bucket. The rate limitation is controlled by the size of the bucket and the leaking hole. A brief explanation of SLB is available at Appendix D. Figure 9 illustrates an ingress, where packets are filtered into classes and policed to appropriate rate and after that sent up to transport-layer.

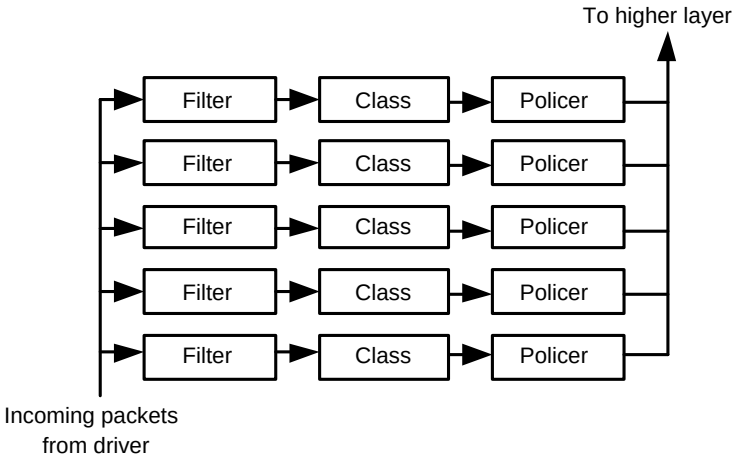

Figure 9: Ingress part of packet control.

The ingress that was built in this project performs filtering of arrival flows into five classes, given in the description below. Packet filtering for the first class have a hierarchical structure. First filtering is based on the link quality value included in the TTL-field and the second filtering is based on the Diffserv value in the TOS-field. This packet class 1, is then policed to the specified rate. For classes two to four, there is u-32 filtering on Diffserv value in the TOS-field. Packets not fitting to a filter are classified as best-effort flow 'class 5' and this packet class is policed to a rate of 1 Mbps. Reason for this shaping is to decrease the amount of this type of traffic at AP.

Class 1. Packets having TTL-value below 223 and above 193.

1.1 Packets having TOS-values: 10, 8 or 4.

1.1.1 SLB to 3 Mbps, burst size of 64 kB and accept minimum packets of 64 Bytes.

Class 2. Packets having TOS-value 10.

Class 3. Packets having TOS-value 8.

Class 4. Packets having TOS-value 4.

Class 5. All other packets.

5.1 SLB to 1 Mbps, burst size of 64 kB and accept minimum packets of 64 Bytes.

5.3.4 Egress

Outgoing packets created at application-layer are forwarded to transport layer and after that to TC-layer in network layer where egress is. Egress is most like ingress; it is a collection of TCNG mechanisms to control packets. Filtering, policing and dropping of packets can be done in the same manner, but a difference from ingress is that packets are queued in separate classes after filtering until transmission. Use of qdiscs and queues to build a packet scheduling structure is only possible at egress, because ingress have only limited set of functions to preliminary classification for policing and drop of undesirable packets. The egress structure that was built for this project at the AP, where the resource sharing takes place is shown in Figure 10. There is a HTB qdisc upon the dsmark qdisc and for each HTB class there is a SFQ queue to store packets before entrance to the scheduler (HTB qdisc will be explained at Section 5.3.5).

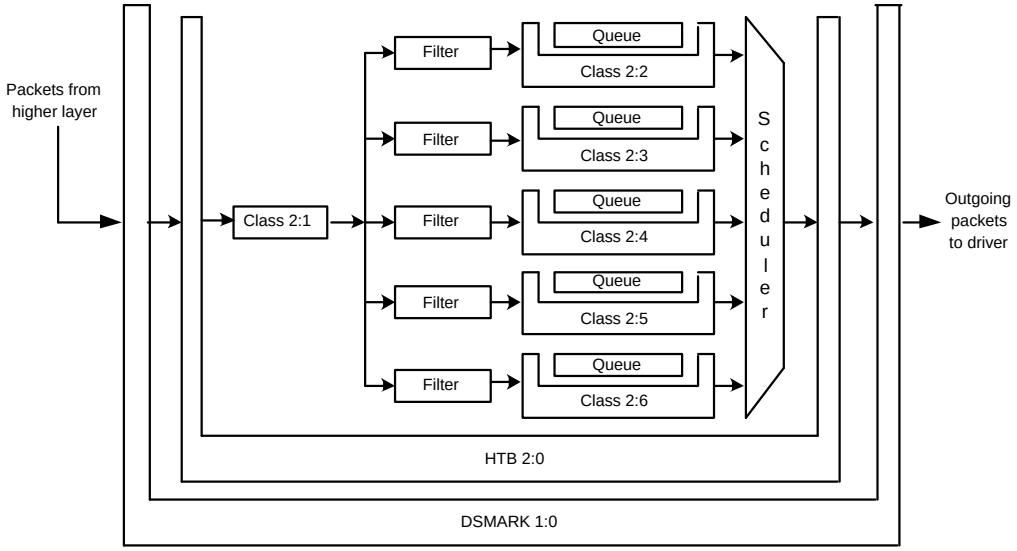

Figure 10: Egress and packet scheduler.

Packets are filtered in the same manner as in ingress, based on the TTL-value and the TOS-value, but instead of policing the packet rate the scheduler takes care of it.

Below is a description of egress shown in Figure: 10, where: the first class filters packets based on the TTL-value range, which corresponds the link quality value. Next filtering is based on the Diffserv definition for delay sensitive and high throughput requiring flows. The second to fourth class are only filtered with u-32 filter group and on the Diffserv values in the TOS-field. Class two will con-

tain packets that are delay sensitive, class three; with throughput constrained flows and class four for applications requiring reliable traffic.

Class 1. Packets having TTL-value below 223 and above 193.

1.1 Packets having TOS-values: 10 or 8.

Class 2. Packets having TOS-value 10.

Class 3. Packets having TOS-value 8.

Class 4. Packets having TOS-value 4.

Class 5. All other packets.

5.3.5 Scheduling

A scheduling policy is a decision of packet serving order in a network node. In an aggregate scheduling at least one packet queue (class) is required and the order of service is dependent on the scheduling policy. Most of existing algorithms for aggregate scheduling are based on parameters that are specifying how resources shall be preferentially treated and shaped. Managing bandwidth among stations in a way that guarantees the requirements can be obtained using a qdisc at egress. The scheduling part in a qdisc is a process of several steps dependent on scheduler. As described in Section 5.3.1, there is a simple scheduler in the FIFO qdisc, that has no classes within it and it has only an optional parameters in queue size limitation. It simply enqueues aggregate flows in arrival order and then dequeues packets in that order to the network card driver. A more advanced scheduler will filter packets into classes, give preferential classes certain amount of resource and special treatment before it shapes the outgoing packet rate using a policier. The HTB qdisc, is a hierarchical link-sharing qdisc that is more advanced than FIFO and takes several specifying parameters. As the name tells, it uses tokens and buckets to serve and to shape traffic for hierarchical class based queues (Figure 10).

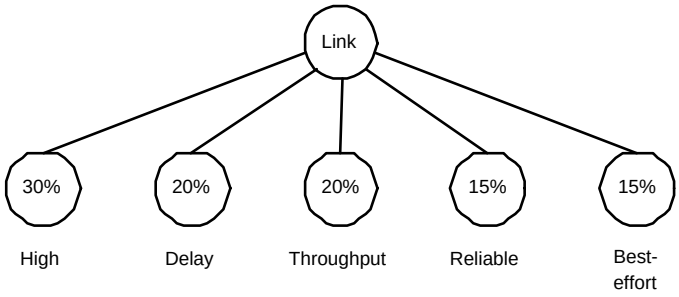

Figure 11: Link sharing structure using HTB.

The HTB qdisc that was used in this project for bandwidth managing has a flat structure where each class has one parent class (Figure 11). Class parameters defined for HTB in TCNG are: rate, ceil and prio. Rate is the only required parameter, ceil and prio are optional parameters. The rate parameter defines the minimum rate that can be allocated from the total bandwidth. Besides the static guaranteed rate of the bandwidth, each class has also a priority that is used for ordered usage of excess bandwidth. The ceil parameter sets the maximum rate that a class can have with borrowed bandwidth. Classes that are specified for guaranteed rate, has the same rate and ceil rate parameter values and are therefore not borrowing exceeding bandwidth. Important is that the sum of all classes' rate shall not exceed the total rate capacity of the parent class [5]. Otherwise will parameters for rate be set by the TCNG to an appropriate rate based on the parent class capacity.

$$Parent\ Class\ capacity > \sum child\ classes\ capacity \tag{5}$$

The HTB configuration for management of the bandwidth in this project was: a parent class 2:0 with maximum bandwidth capacity of 5.5 Mbps (Figure 10). Child classes with rate, ceil and prio configurations were created for each filtered class of packets. Named classes in the scheduler part are configurations for the scheduler and are matched to classes created at the filtering. A description of the construction is shown below in Figure 12. Class one consists of packets described in Section 5.3.4 and at the scheduler part this class is matched to a class configured with the rate: 1,650 Mbps, ceil: 5,5 Mbps and lowest prio value: 0 .

[5]Traffic Control, http://www.tldp.org/HOWTO/Traffic-Control-HOWTO/index.html (Visited 2004-02-08)

```
/*-- Outgoing interface ---*/
egress {
    /*-- Filtering --------*/
    class (<$1>) if ($h && ($DS10 || $DS08));
    class (<$2>) if $DS10;
    class (<$3>) if $DS08;
    class (<$4>) if $DS04;
    class (<$5>) if 1;
    /*-- Scheduler -------*/
    htb {
        class (rate 5.5Mbps, ceil 5.5Mbps){
        $1 = class (prio 0, rate 1.650Mbps, ceil 5.500Mbps){sfq;}
        $2 = class (prio 1, rate 1.100Mbps, ceil 5.500Mbps){sfq;}
        $3 = class (prio 1, rate 1.100Mbps, ceil 5.500Mbps){sfq;}
        $4 = class (prio 3, rate 0.820Mbps, ceil 0.820Mbps){sfq;}
        $5 = class (prio 2, rate 0.820Mbps, ceil 3.000Mbps){sfq;}
    }
}
```

Figure 12: Class description of a HTB scheduler

There is a high flexibility in qdisc and queue combinations when composing packet scheduler in Linux. The decision to use SFQ was that a class can contain packets from several flows and the service will be fair distributed among represented flows in the queue [21]. In a scenario where a class is dominated by one flow, the service in the queue among flows will still be fair.

5.4 Channel Monitor

Complex QoS system can be built to manage the bandwidth and schedule network traffic by using TC mechanisms, but as Lars Wischhof explains in [32]: traffic control is not developed for wireless network, neither is the IP-protocol. There is a lack of link condition field in the IP-header and there is lack of a channel monitor in TC. Beside the traffic controlling with queues and filtering, there is also a need to log- and bit setting (mangling) packet header fields. The log function is needed by the monitor to store packet flows TTL-value on uplink and to set back the TTL-value for outgoing packets. Marking is needed to set an appropriate TOS-value for packets and these bit settings of packet headers have to happen before entering egress where the scheduler is. Logging and mangling packets in the IP-layer can

be done using the Linux kernel security mechanism netfilter [6] . Netfilter is used for: filtering, mangling and masquerading IP-packet at a network node. The filtering mechanism differs from the one explained for TC and used in tcng. Packets can be filtered dependent on IP-header fields, protocol, source/destination address and more, but filtering is for either processing or discarding packets. Mangling is like the marking option in TC, but tcng can only mark packets dependent on TOS-field value, Netfilter can set: TOS, TTL and mark packets. This tool has two parts: Netfilter is the kernel side; and Iptables is the user space tool. Netfilter options can be used from user space using the Iptable[7] tool, which has an interface to the kernel-space modules. Iptables tool consists of commands that can be executed at user space or several commands included in a script file. Iptable commands used in this project are for: filtering, logging, marking and mangling. Targets of executed commands are for tables within a network chain in the kernel, where the packet operation takes place. Network chains that are used in this project are:

- **INPUT table**
  Packets to specified ports are logged every second. The logging mechanism has a limitation option configured to log specified number of packet headers at every logging moment. This information will be used to monitor link condition. The time interval for storing and the number of packets to store at each time are empirical calculated based on the bandwidth rate and necessary logging.

- **OUTPUT table**
  Packets that are leaving the specified source ports of the AP, are mangled in the TOS-field with values: 10, 8, and 4.

- **POSTROUTING table**
  This is the last chain before packets leaves the AP. In this case, the monitor uses it to set the TTL-value that was delivered in the uplink.

The monitor is a stand-alone process running in the kernel with the purpose to set the TTL-value that flows had on the uplink into corresponding flows on the downlink. Quality value setting for specified outgoing packet flows will follow the procedure explained in Figure 13. Netfilter will log specified flows TTL-value at the INPUT chain for incoming traffic to AP. Outgoing flows passing the OUTPUT chain will have TOS-field set with predefined values only if the chain has a command for this. The monitor will take the last log of TTL-value for

[6] Netfilter kernel modules come with Linux kernel-2.4.20

[7] Iptables Tutorial 1.1.19, http://iptables-tutorial.frozentux.net (Visited 2004-02-23)

specified flows and create a new Iptable command. The Iptable command that sets the TTL-value at mangle table in POSTROUTING chain will be replaced with this new created command. This process of setting the TTL-value will be done every third second for specified flows only if link condition value has been changed. Filters at the HTB qdisc will then classify packets dependent on the TOS-value given at OUTPUT-chain and the TTL-value set at POSTROUTING chain[8].

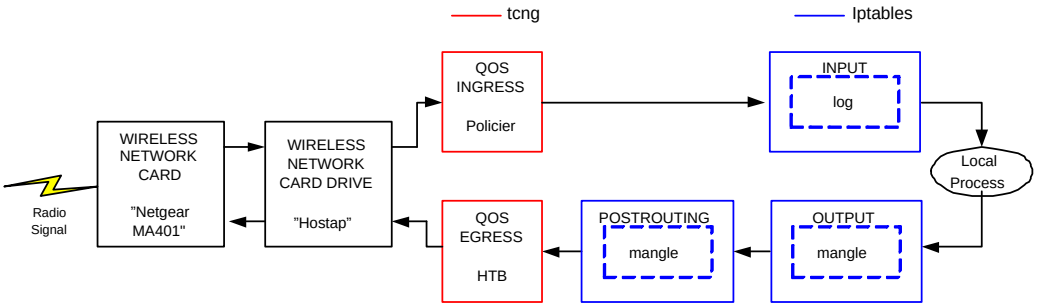

Figure 13: Control and filtering of packets.

[8]Iptables Tutorial 1.1.19, http://iptables-tutorial.frozentux.net (Visited 2004-02-23)

6 Design of test environment

The evaluation of the packet scheduling in a wireless network using HTB was done in three tests. Three stations were used in test scenarios and configurations of those are described in Table 2. Specifications for each test are described in each sub section. Configurations that were the same for all three tests:

- Packets transmitted from AP on the downlink to stations were backlogged using the packet generating application Iperf [9].
- Iperf is used as server and client, where the sender is server and the receiver is client. All stations and AP were both server and client since there was traffic generated in both directions.
- Downlink packets size were set to the default value: 1470 Bytes at Iperf for all flows. Reason to use same packet size instead of randomizes as in a reality, was to have regularity in tests and to have comparable results.
- Packet-generating rate were configured at Iperf to 11 Mbps for all flow in both up- and downlink. This rate is totally different from the bandwidth rate configured for wireless network cards.
- The transmission rate was set to 11 Mbps at all stations and AP:s network cards. Preparation tests that were done before these tests showed that the bandwidth rate capacity was 5,5 Mbps.
- Positions and distances between stations and AP during tests are shown in Appendix A.
- The packet transfers in the test scenarios were logged at each station and AP, for both up- and downlink traffic. The tcpdump application was chosen to log packet because of its high flexibility. tcpdump is a user space application that makes an own copy of all packet headers entering and exiting the monitoring station. Logging of incoming flows took place before the PREROUTING chain and of outgoing flows after the POSTROUTING chain (see Appendix E).
- QoS parameters that were explained in Section 4.2: bandwidth, jitter and losses is measured and logged by Iperf.

[9] Iperf Version 1.7.0, http://dast.nlanr.net/Projects/Iperf/ (Visited 2004-02-23)

	Station Leela	**Station Fry**	**Station Amy**
Hardware	Dell	Dell	IBM
Network card	Netgear Ma401	Orinoco Gold	Orinoco Gold
IP-number	192.168.100.100	192.168.100.101	192.168.100.102
Linux Kernel	2.4.20	2.4.20	2.4.20
File system	ext3	ext3	ext3
Wireless unit	AP	Station (STA1)	Station (STA2)
Wireless drive	hostap-0.1.2	orinoco-piggyb.	orinoco-piggyb.
Wireless extension	14, 15	14, 15	14, 15
Wireless mode:	Master	Managed	Managed
Encryption:	Off	Off	Off
Rate:	11Mbit/s	Auto	Auto
Traffic control	tcng 9f		
Netfilter	Iptables 1.2.8	Iptables 1.2.8	Iptables 1.2.8
Packet generator	Iperf 1.7.0	Iperf 1.7.0	Iperf 1.7.0
Traffic logger	tcpdump	tcpdump	tcpdump

Table 2: Configurations on test machines

6.1 Fairness test

Purpose with this test was to see how results from this test differs from a test with a scheduler and also to see how the bandwidth is shared equally among six flows with only downlink traffic (Figure 14).
AP had three flows to each other of the two stations. Flow number two was started three seconds after the first one. Flow three and four: started at same time after three elapsed seconds from start of flow number two. After another three seconds flow five, and after that flow six was started.

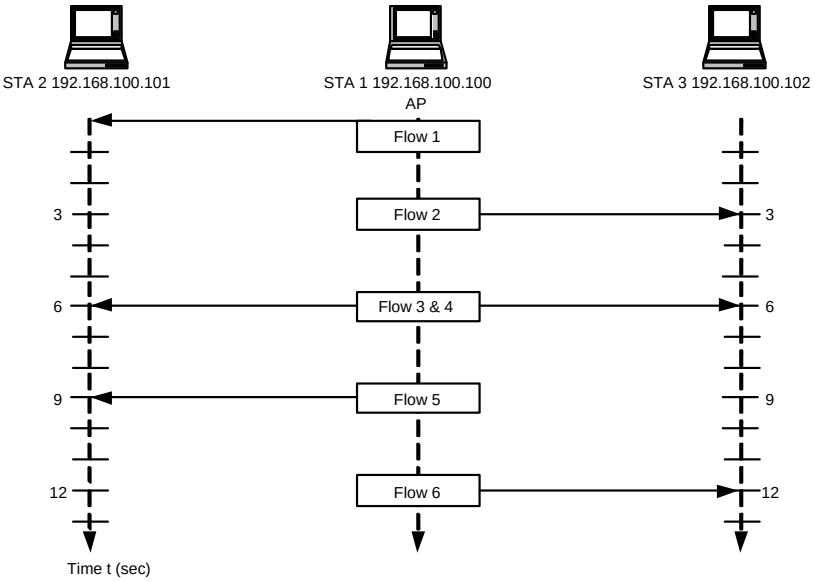

Figure 14: Fairness test scenario.

6.2 Scheduler test

The aim with this test was to measure the performance of the HTB scheduler and check that packet classes got the assigned bandwidth quantity that was set in the scheduler. This test was also for comparison of service curves with the fairness test to see if there were any improvements of traffic flows with scheduling and to measure QoS parameters.

- The scheduling algorithm HTB were loaded into the Linux kernel using a TC script.
- Packets were generated from AP as in the first test (Figure 14).
- The scheduler structure was built as the description in Section 5.3.5 and configured to share the bandwidth capacity as in Figure 15.

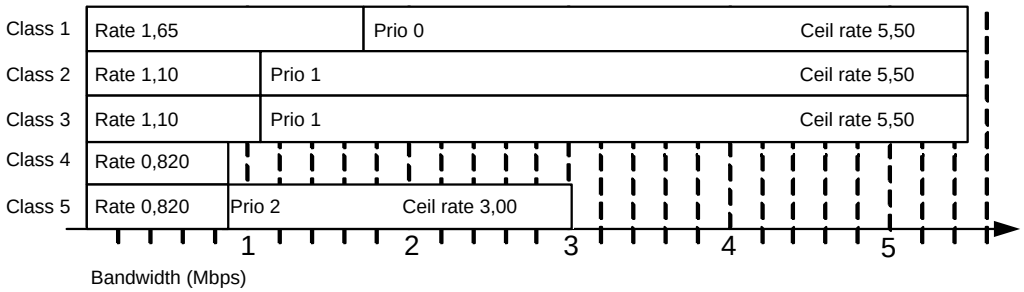

Figure 15: Class based link sharing.

6.3 Scheduling test with quality value

This test is for measuring the performance of the downlink scheduler based on TTL-value stations had on the uplink. Besides this, the aim was to see if there would be any improvements on QoS parameters when scheduling with signal quality values.

Start order of flows in this test was the same as in the fairness and scheduling tests. The difference from the scheduling test is that: STA1 and STA2 are transmitting packets on the uplink with link quality value in the TTL-field to AP (Figure 16). STA2 started transmitting uplink traffic one second before AP started its downlink flows. Reason for this was to provide AP with current TTL-values, for those flows that were scheduled based on signal quality values.

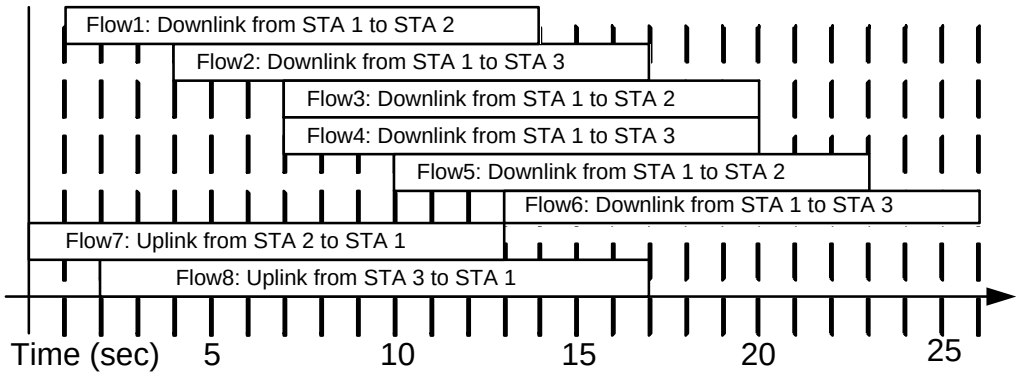

Figure 16: Flow start order.

The medium access rights for AP to transmit its downlink traffic are the same as for a station in the BSS. AP is one among all contending stations for transmission in the medium and it also uses the MAC algorithm CSMA/CA. This makes it difficult to do a fair measurement of traffic scheduling on the downlink with a quality value. It is more complicated when the AP cannot send and receive packets at the same time, since the communication is half duplex. To not affect the performance of the downlink scheduling too much, different size of packets were used for up- and downlink. Smaller packets transmitted on the uplink will delay AP for a short time and more time will be for contending/transmission on the downlink. Packet size for downlink flows was set to default value: 1470 Bytes as there were in fairness and scheduling tests. For uplink traffic from STA1 and STA2 packet size configuration was 128 Bytes at Iperf.

7 Test results

7.1 Fairness results

Figure 17 shows service curves from the fairness test, where bandwidth capacity was shared equally among all flows. For each started flow at AP, available downlink capacity was divided with 1/N (N is the number of flows). When the time period for a flow is ending its allocated rate is shared among remaining flows, as it was supposed to do by the definition of IEEE802.11b standard.

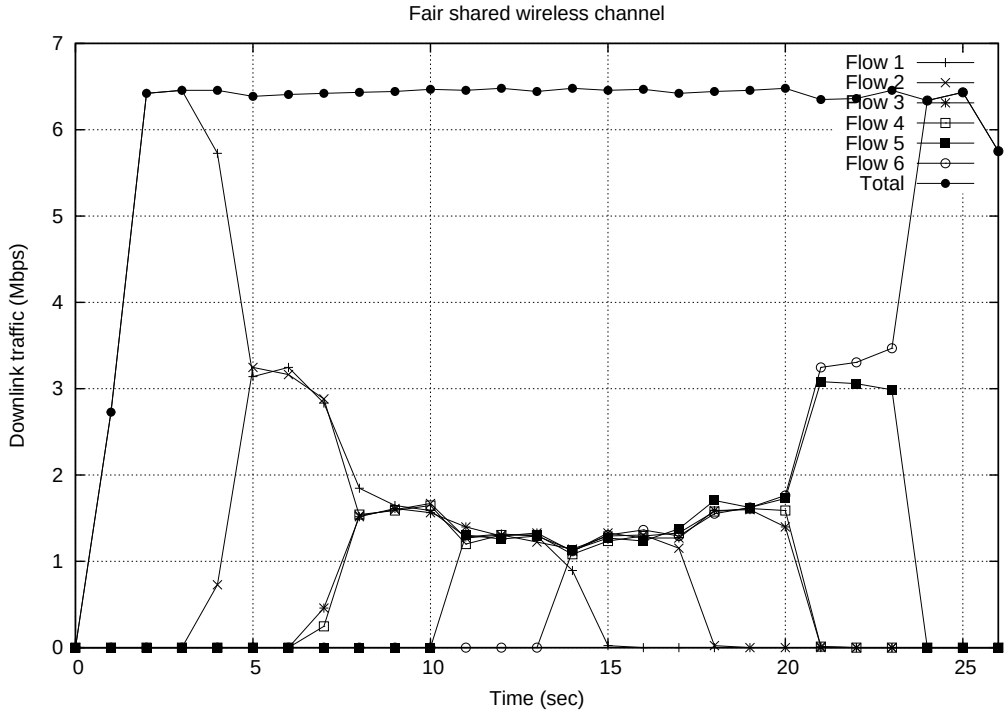

Figure 17: Fairness test results.

7.2 Scheduling results

Service curves from this test show that the bandwidth capacity was divided as the HTB was configured to do (Figure 18). The first half of the test period shows that flow one with the highest priority for exceeded bandwidth also utilized most of the capacity.

In comparison with the fairness test, during the time period from the eighth to 12th second, flow one had a throughput of average 1,5 Mbps. During this test and the same time period flow one had average 2,4 Mbps. Flow one received 0,9 Mbps more bandwidth in scheduled traffic though the total bandwidth capacity was approximately 1 Mbps less in this test.

Flow two was classified as normal traffic though station STA1 sent packets with TTL-value of 213. This was because of the scheduler's configuration to classify flows with TTL-value and TOS-value set at the same time. Only a TTL-value set is classified as normal traffic. From the start of flow two; tenth second, it gets what a normal class flow shall receive: 0,820 Mbps. After the tenth second flow five starts and this is also a normal traffic flow, so flow two and five two share the total capacity for normal traffic. Priority value for those two flows is prio: 2, but cannot receive any exceed capacity, since there are flows having higher priority.

It is also shown from the graph that: during the time period: 12-15th second, where flow one is decreasing; flow three and four are taking over exceeding bandwidth. Flow three and four both with priority value one, increases their capacity. Flow three belongs to class three and flow four to class two, so flow four receives more than flow three.

Flow six was classified to class four, which was configured at scheduler for rate: 0,820 Mbps and ceil rate: 0,820 Mbps. This flow gets this rate and keeps this capacity during its lifetime.

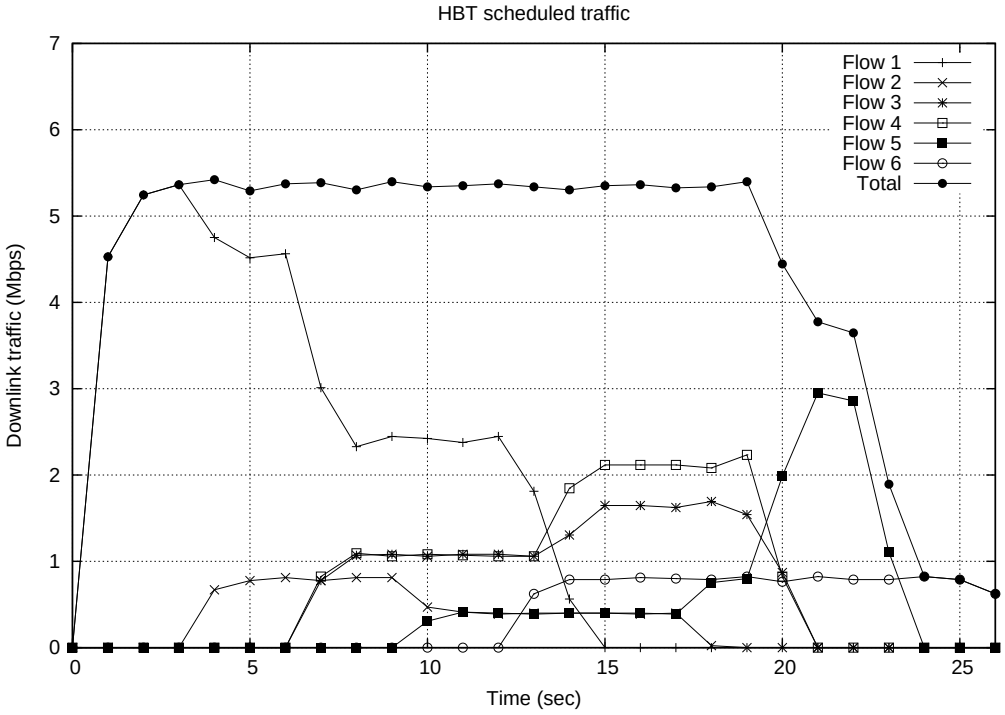

Figure 18: Scheduling test results.

7.3 Scheduling with quality value results

Results from this test shows that the total downlink capacity is less than the scheduling capacity (Figure 19). Reason for this is the increased traffic with both uplink traffic from stations and downlink traffic from AP. AP cannot be configured to transmit more packets on the downlink than on the uplink, as it was described at Section: 6.3. Packets transmitted on the uplink with smaller packet size took shorter time than downlink packets. More time were for transmission on downlink, but the overall capacity loss during the test time was approximately 1,5 Mbps and depends on this matter.

The uplink traffic is affecting all flows on the downlink. In comparison with the scheduling test with smooth service curves for each flow, this test results shows serrated curves when AP altering between receiving and sending. In the time period between from start to third second; STA2 has an uplink and flow one a downlink a second later. When STA3 starts to compete for the uplink at third second the capacity for flow one decreases with approximately 1 Mbps. This is the first spike shown for flow one and illustrates how much contention means. When STA2 and

STA3 stops their uplink flows at 17th and 20th second, remaining downlink flows in the test: flow five and six are behaving most like in the scheduling test. Also the total downlink capacity is increased to values that were achieved in the scheduling test.

The log-file of TTL-values collected at the netfilter chain; INPUT, shows that link quality value can change for a station with a static position. This changes depends on multi path propagation of the signal and since the signal quality value was not below 193, flow one could not alter from class one to class two in the scheduling.

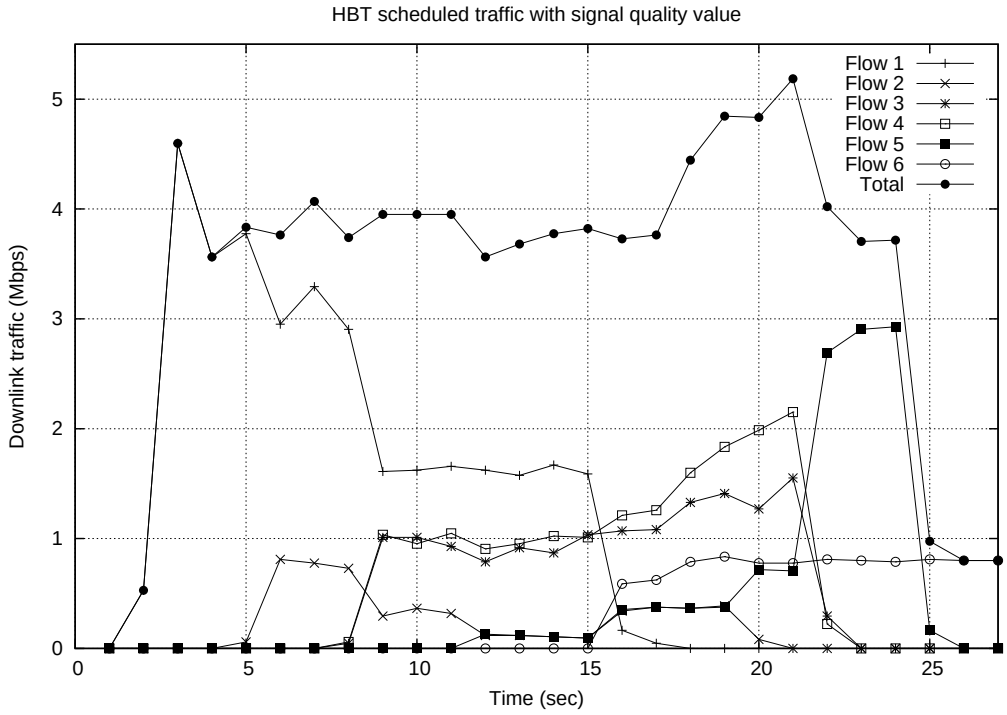

Figure 19: Scheduling with quality value results.

7.4 Scheduling performance and QoS results

Results from the second and third test show that the HBT scheduling algorithm performs well. Performance in the meaning that it meets all the requirements of managing the downlink in a hierarchical scale and classifying flows to class based share of the capacity. Test results showing this have been presented in earlier test results and a comparison graph for all three test is presented in Figure 20.

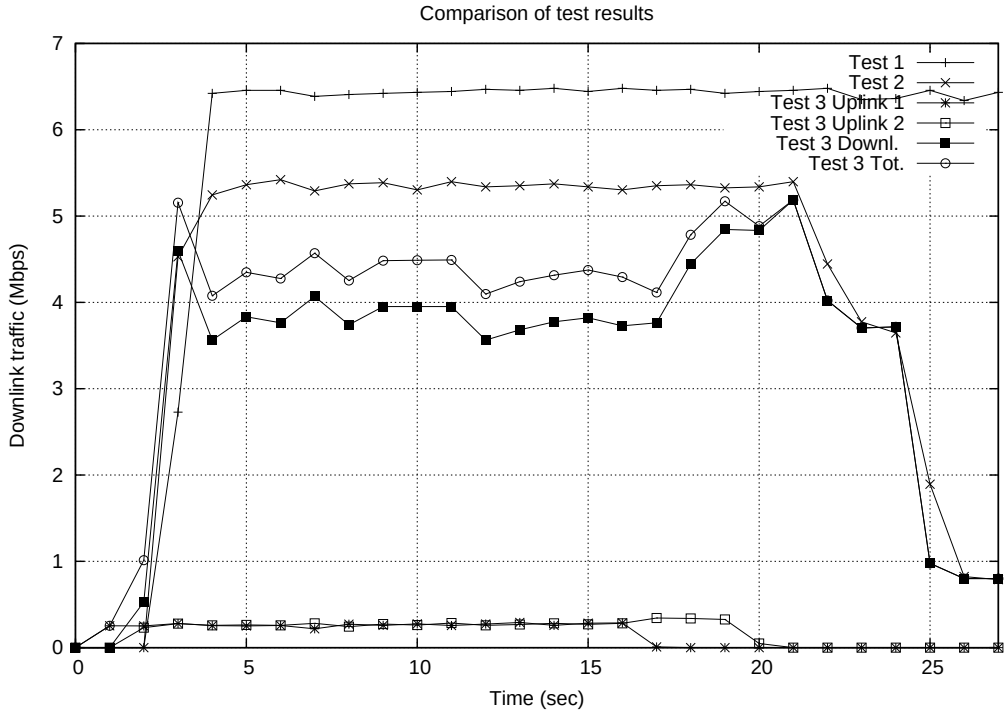

Figure 20: Comparison of three tests.

Results for implemented QoS with differentiated service parameters (Section 4.2) were investigated. Stations in the test had to be synchronised at both sender and receiver for accurate results in measurement of jitter and latency. This synchronisation seemed to be difficult to achieve on those machines used in the tests. A short time after synchronisation, the time differences of stations were more than one second, which is to big for being accurate. Time difference of stations clocks cannot be that big, since jitter values are of mille- second scale. Instead of measuring jitter with synchronised stations, the summarised information given by Iperf containing the jitter value was used.

Table 4 at Appendix B shows flow specific results for all flows measured by Iperf. This table provides QoS parameter values to estimate the QoS performance:

- **Bandwidth**
  Packets were filtered and classified into classes. Classified packet flows received the predefined bandwidth in the scheduler, which test results from test two and three showed.

- **Latency**
  This parameter was rather difficult to measure without synchronisation and without a measurement from sending process to receiving process. The jitter value describing latency variation between arrivals of packets was used instead.

- **Jitter**
  Jitter results are illustrated in Figure 21 and showed that all flows except: flow two and six got either better or less changing jitter values. Reason for increasing jitter values was that these two flows got less bandwidth capacity in test two and three than in the first fairness test. Flow one got more bandwidth in test two and three and hence got better results. Remaining flows got a bandwidth nearly the same what they got in the fairness test and therefore was the change not remarkable.
  There are small changes in jitter values for all flows between test two and three and this difference is to small to say that it depends on the uplink traffic that was introduced in test three but not in two.

- **Losses**
  With scheduling of the downlink traffic, all flows got less or no loss at all during test periods (Table 4).

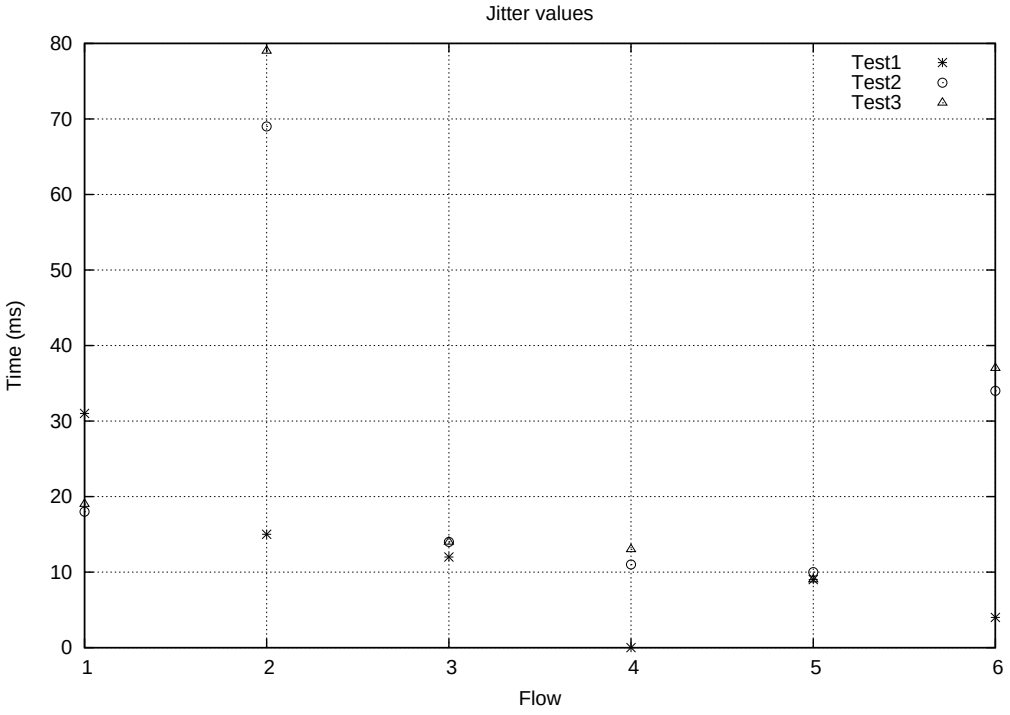

Figure 21: Jitter values measured during tests.

8 Discussion

The purpose of this project was to investigate possibilities for packet scheduling on the downlink in a wireless network. If it is possible, then packet flows will be prioritised in the downlink with a scheduler dependent on link quality each station perceives. Intention to scheduling flows dependent on the link quality came with the lack and demand to control sharing of the downlink capacity in a wireless network using the IEEE802.11b standard. The standard takes no notice to the radio signal quality that stations receive from the AP. The link capacity is shared equally among all flows and no preferential treatments are for those stations with good link quality. The HTB scheduler was used for sharing the downlink among flows dependent on link quality values to their destinations. This scheduler serves packet flows that are classified into hierarchical structures, where each class is guaranteed a specific minimum quantum of the bandwidth capacity. Decision for usage of this algorithm was the hierarchical link sharing that suits the decision to schedule downlink capacity based on the link quality between stations and AP. The simplicity in usage and performance results presented in earlier tests based on wired network were convincing to test it in wireless network.

Every station in the wireless network measured the radio signal strength from their associated AP and calculated a quality value representing the link quality. This link quality parameter was sent to AP in the TTL-field of the IP-packet header (Figure 22). Reason to use already existing field, was that if this value had been included as a piggyback value into the packets body, it should been additional processing to recalculate the packets checksum after inclusion of the parameter and also do the reversed process at the receiver. This should not be a realistic scenario and it should take more process and effort than needed.

The link quality values from stations were stored in a monitor at AP. Reason to use a monitor was that the link quality information comes on the uplink and it is needed in outgoing packets on the downlink. There is no link between up- and downlink network stacks or a mechanism to deliver this quality value to downlink flows where this information is needed. Another difficulty arises at the scheduler, as it cannot store station specific link quality value and it cannot be configured on the run. The Netfilter tool was therefore used to set back the current quality value supported by the monitor into corresponding packet header of the flow. This mangling of packet header happened before packets entered the scheduler and were classified based on this quality value.

The packet scheduler's configuration for link sharing had no scientific basis during tests. Important in settings for link-sharing parameters was that traffic flow

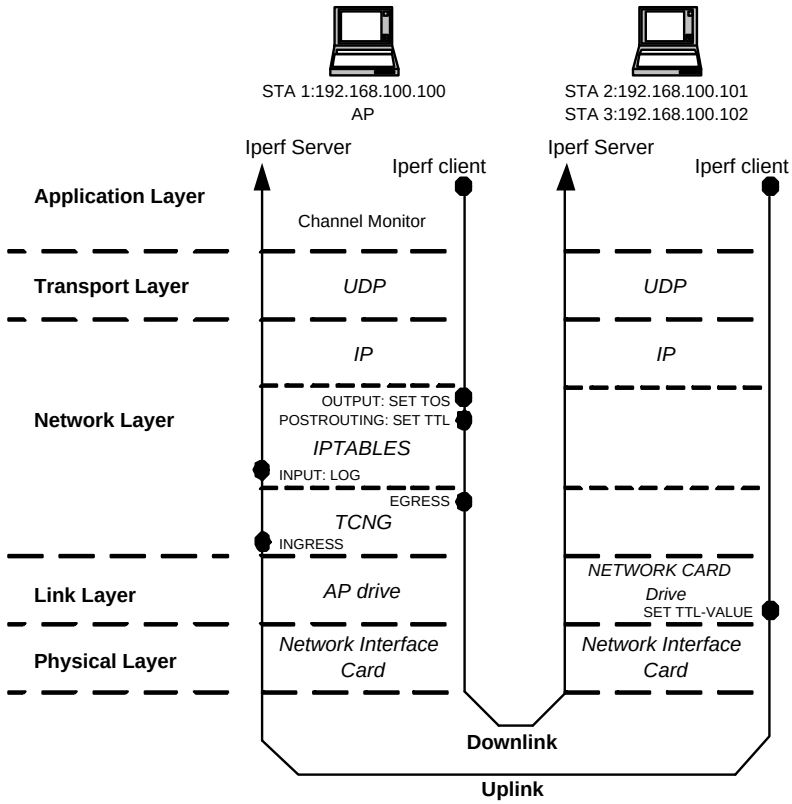

Figure 22: Big picture of scheduling design.

class's aggregate capacity should be less than the available bandwidth capacity. Tests were done for performance measurements on the scheduler. During these tests, packets were sent with the TOS-field of the packet header set to differential service values. The type of service field describes the required service from the application and it was used to attain high level of QoS. The only common parameter for all generated packet flows were the same packet size. Instead of using a randomised packet size, which had been more realistic, the static value was for comparison of QoS parameters like jitter and latency among all flows. A more realistic scenario would have been using 512Byte packet size for delay sensitive packets as there is in Voice applications. This should give realistic jitter values.

Five principles that were described in Section 4 to achieve the QoS, is mostly fulfilled in the solution for this project.

- The "Integration principle" stated that the service should be configurable in all IP architectural layers to obtain quality of service in an end-to-end transfer. This is done at the physical layer by controlling the packet flow transmission using a scheduler from the TC-layer in the network-layer. At data-link-layer, packet headers were set for delivering the quality value in the TTL-field. At network layer; traffic were controlled and filtered for packet flow classification in purpose to serve the scheduler. At Transport layer nothing were done, but it was stated in tests that packet size is in relation to transmission time. The packet size and type have to be adequate for the traffic generated from the application.
- "Separation principle" stated that: classification of packets for prioritising flows dependent on required service. Tests have showed that this is possible.
- "Transparency principle": Packets are generated by an application (Iperf in tests) and at layer below, transparent to the application, treated for QoS with: classification, policing, queuing and scheduling. Theses operations are independent and free from the application. Interserv is not transparent, when there is packets sent for signalling resource reservation.
- "Asynchronous resource management principle": This was what the HTB scheduler did with downlink flows. Flows were controlled and bandwidth was managed to classes asynchronously. Exceeded bandwidth were shared to classes.
- "Performance principle": Results from tests showed that; in comparison with a best-effort service, scheduled flows which were filtered and classified, created order in the meaning that packet loss rate decreased (see Appendix B) and flows got specific bandwidth.

Results from tests show good performance for both the scheduler and QoS. The downlink capacity decreased a little when the scheduler was used and even more when there where both up- and downlink traffic. Seemed to those QoS parameters, the advance in using a packet scheduler, scheduling dependent on the quality value is better than best-effort traffic. There is a reservation for QoS parameter results. These results were measured at stations and factors' affecting these results are wireless PCMCIA network cards limitations in transmit range and transmit buffer size. Other factors is the PCMCIA cards antenna diversity and indoor propagation. Results can be revaluated with these factors considered.

This end-to-end design with scheduling dependent on link quality value and improvement of QoS is not only for tests, it can also be used in reality. The differential service codes set in the TOS-field of a packet can be used as a parameter to the scheduler alone instead of the modified network card driver measuring and calculating the link quality along with TOS-fields. When the modified Orinoco driver is used and in a scenario where packets are leaving the domain, the TTL-value has to be set to default value or recalculated at AP.

The main reason to design this project in Linux environment was the stability and robustness the operating system has shown in tests before [10]. Besides this, Linux kernel-2.4.20 comes with an environment for development; including programming languages like C, C++, Perl and Bash-shell to interpret user-defined commands in script files [22]. This project became a solution of blocks, where each block consists of a Linux tool. Blocks that made it possible to build the designed solution were: networking facility in TC and netfilter, the open source developed Host-AP [11]application by Jouni Malinen for wireless communication, the Orinoco network card drive to measure, estimate and include the link-quality value into the TTL-field. To make these blocks work together interfaces were built between them using bash scripts.

The purpose of this project was to study: the feasibility and performance of an algorithm, which prioritises traffic from stations with good link quality to an optimal operating rate and distributes the "exceeding" bandwidth evenly among stations across the network. This solution shows balanced traffic distribution in the BSS though prioritising, and this also gave minimized queuing delay and losses.

[10]Kernel 2.4.20, http://www.kernel.org (Visited 2004-02-23)
[11]Hostap-0.1.2, http://hostap.epitest.fi (Visited 2004-02-23)

References

[1] ALMESBERGER, W. Linux Network Traffic Control - Iplementation Overview. EPLICA, February 2001. http://www.almesberger.net/cv/papers.html (Visited 2004-02-23).

[2] BHARGHAVAN, V., LU, S., AND NANDAGOPAL, T. Fair Queuing in wireless Networks: Issues and Approaches. In *IEEE Personal Communications* (University of ILLINOIS, February 1999).

[3] BIRMÉ, J., FAHLGREN, D., SÖDERMAN, G., REVEMAN, D., AND ÅSLUND, F. Linux TC. Tech. rep., Department of Computer Science, Umeå University, May 2003. Project Linux TC.

[4] BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z., AND WEISS, W. An Architecture for Differentiated Services. Network Working Group, December 1998. RFC2475.

[5] BOYLESTAD, R. L. *Introductory Circuit Analysis*. Prentice Hall International Inc, New Jersey, 2000, ch. 21, pp. 911–915.

[6] BRADEN, R., CLARK, D., AND SHENKER, S. Integrated Service in the Internet Architecture: an Overview. Network Working Group, July 1994. RFC1633.

[7] BRADNER, S. Latency key in wireless-net management. Network Working Group, July 1991. RFC1242.

[8] CAMPBELL, A. T. A Quality of Service Architecture. Tech. rep., Computing Department Lancaster University, January 1996.

[9] CIMEN, J. Power Awareness in Wireless Mobile Ad Hoc Network. In *Proceedings of Umeå Student Conference in Computer USCCS03* (Umeå University, June 2003), J. Börstler and K. Sullivan, Eds., vol. 3, Department of Computer Science, Umeå University, pp. 107–116.

[10] COULARIS, G., DOLLIMORE, J., AND KINDBERG, T. *Distributed systems concepts and design.* Addison-Wesley, 2001, ch. 15, pp. 614–627.

[11] CROW, B., WIDJAJA, I., KIM, J., AND SAKAI, P. IEEE802.11 Wireless Local Area Networks. *IEEE Communication Magazine* (September 1997), 116–126.

[12] DAVIE, B., CHARNY, A., BENNETT, J., BENSON, K., BOUDEC, J. L., COURTNEY, W., DAVARI, S., FIROIU, V., AND STILIADIS, D. An Expedited Forwarding PHB (Per-Hop Behavior). Network Working Group, March 2002. RFC3246.

[13] DEFENSE ADVANCED RESEARCH PROJECTS AGENCY INFORMATION PROCESSING TECHNIQUES OFFICE. Darpa Internet Program Protocol Specification. Information Sciences Institute University of Southern California, September 1981. RFC791.

[14] DEMICHELIS, C., AND CHIMENTO, P. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). Network Working Group, November 2002. RFC3393.

[15] ECKHARDT, D. A., AND STEENKISTE, P. Improving Wireless LAN Performance via Adaptive Local Error Control. *Proceedings of IEEE ICNP '98* (1998).

[16] ELMASRI, R., AND NAVATHE, S. B. *Fundamental of Database Systems*. Addison-Wesley, 2000, ch. 9, pp. 308–311.

[17] HEINANEN, J. Assured Forwarding PHB Group. Network Working Group, June 1999. RFC2597.

[18] HUSTON, G. Next Steps for the IP QoS Architecture. Network Working Group, November 2000. RFC2212.

[19] IEEE 802.11 WORKING GROUP. Draft Standard IEEE802.11 Wireless LANs. The web, June 1999.

[20] KUROSE, J. F., AND ROSS, K. W. *Computer Networking*, 1 ed. Addison-Wesley, Boston, 2002.

[21] MCKENNEY, P. Stochastic Fairness Queuing. IEEE INFOCOM, June 1990.

[22] MITCHELL, M., OLDHAM, J., AND SAMUEL, A. *Advanced Linux Programming*, 1 ed. New Riders Publishing, New Jersey, June 2001, ch. 1, pp. 10–11. http://www.codesourcery.com/publications.html (Visited 2004-02-23).

[23] NANDAGOPAL, T., AND GOA, X. Fair scheduling in Wireless PAcket Data Networks. In *Handbook of wireless networks and mobile computing* (New York, 2002), S. Ivan, Ed., John Wiley and sons, Inc, pp. 171–191.

[24] NICHOLS, K., BLAKE, S., BAKER, F., AND BLACK, D. Defenition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Network Working Group, December 1998. RFC2474.

[25] RADHADKRISHNAN, S. Linux - Advanced Networking Overview. Tech. Rep. 1, Information and Telecommunications Technology Center, Department of Electrical Engineering & Computer Science, The University of Kansas, August 1999.

[26] R.BRADEN, ZHANG, L., BERSON, S., HERZOG, S., AND JAMIN, S. Resource ReSerVation Protocol, Version 1 Functional Specification. Network Working Group, September 1997. RFC2205.

[27] RODOPLU, V., AND MENG, T. H. Minimum energy mobile wireless networks. In *IEEE International Conference on Communications, ICC'98* (Atlanta, 1998), pp. 1633–1639.

[28] SANDALIDIS, H. G., AND STAVROULAKIS, P. Hesuristics for Solving Fixed-Channel Assignment Problem. In *Handbook of Wireless Networks and Mobile Computing* (New York, 2002), I. Stojmenovic, Ed., Wiley & Sons, INC., pp. 51–70.

[29] SCHILLER, J. H. *Mobile communications,* 2 ed. Addison-Wesley, London, 2003.

[30] SHENKER, S., PARTRIDGE, C., AND GUERIN, R. Specification of Guaranteed Quality of Service. Network Working Group, September 1997. RFC2212.

[31] STALLINGS, W. *Data & Computer Communications*, 6 ed. Prentice Hall International Inc, New Jersey, 2000, ch. 3, pp. 89–100.

[32] WISCHHOF, L., AND LOCKWOOD, J. Packet Scheduling for Link-Sharing and Quality of Service Support in Wireless Local Area Networks. Master's thesis, Applied Research Laboratory, Washington University in St. Louis, November 2001. WUCS-01-35.

[33] WROCLAWSKI, J. The Use of RSVP with IETF Integrated Services. Network Working Group, September 1997. RFC2210.

Glossary and abbreviations

AF	Assured Forwarding
Backlogged	The Queue of accumulated packets that are going to be processed or transmitted is never empty.
Backoff	A randomised value a station counts down before a new attempt to access the wireless medium.
Bandwidth	Maximal data transfer rate between sender and receiver.
BER	Bit Error Rate
BSS	Basic Service Set
CA	Collision Avoidance
Class	A queue containing classified packets is called classless, since it does not contain other classes.
Classful	A class containing other classes.
CSMA	Carrier Sense Multiple Access
Datagram	Packets are treated independently, without reference to previous or packets after.
DCF	Distributed Coordination Function
Diffserv	Differentiated Service, offering of services in computer communication without signalling and is a packet class definition
DIFS	DCF Inter Frame Spacing
DSMARK	Differentiated Service MARK
EF	Expedited Forwarding
Egress	A queuing discipline to build traffic controlling modules like: filtering, policing and scheduling on for outgoing packets.
FIFO	First In First Out queue.
Filter	One or several; rules or conditions to match/classify packets into classes.

HOL	Head of a Line
HTB	Hierarchy Token Bucket
IEEE802.11b	Institute of Electrical and Electronics Engineers, 802.11b is the Wireless LAN standard.
IETF	The Internet Engineering Task Force
Ingress	A queuing discipline to build traffic controlling modules like: filtering and policing on for incoming packets.
Interserv	Integrated Service, an architecture that guarantees a quality of given service after a service level agreement using packet signalling.
Iperf	Packet generating application.
Iptables	User space tool of Linux security mechanism Netfilter, for packet filtering in Linux.
Jitter	The time variation between packet arrival, caused by: network congestion, route changes or physical devices timing
MAC	Media Access Control
Mangle	Bit setting in fields of a IP-packet header.
Netfilter	Packet filtering commands included into the Linux kernel to filter, control and drop packets.
PHB	Per Hop Behaviour
Policing	A process of handling out packet flows into a traffic profile. Packets not fitting pattern are discarded.
prio	Priority, optional parameter for HTB classes.
qdisc	Queueing discipline, An algorithm that manages packets in a network device.
QoS	Quality of Service, using principles to provide a quality of service in a computer network
RSVP	Resource Reservation Setup Protocol
Shape	A process in TC to conform packets into a defined traffic profile.

SIFS	Short Inter Frame Spacing
Simplex	Communication in one direction only, also called half-duplex.
SLB	Single Leaky Bucket
SNR	Signal to Noise Ratio
TBF	Token Bucket filter
TC	Traffic Control
TCNG	Traffic Control New Generation
tcpdump	Packet logging application for packet switching networks.
u32	Classification based on the header fields within an IP-packet.
UDP	User Datagram Packet

A Test area

STA1 is the AP, STA2 and STA3 are stations.

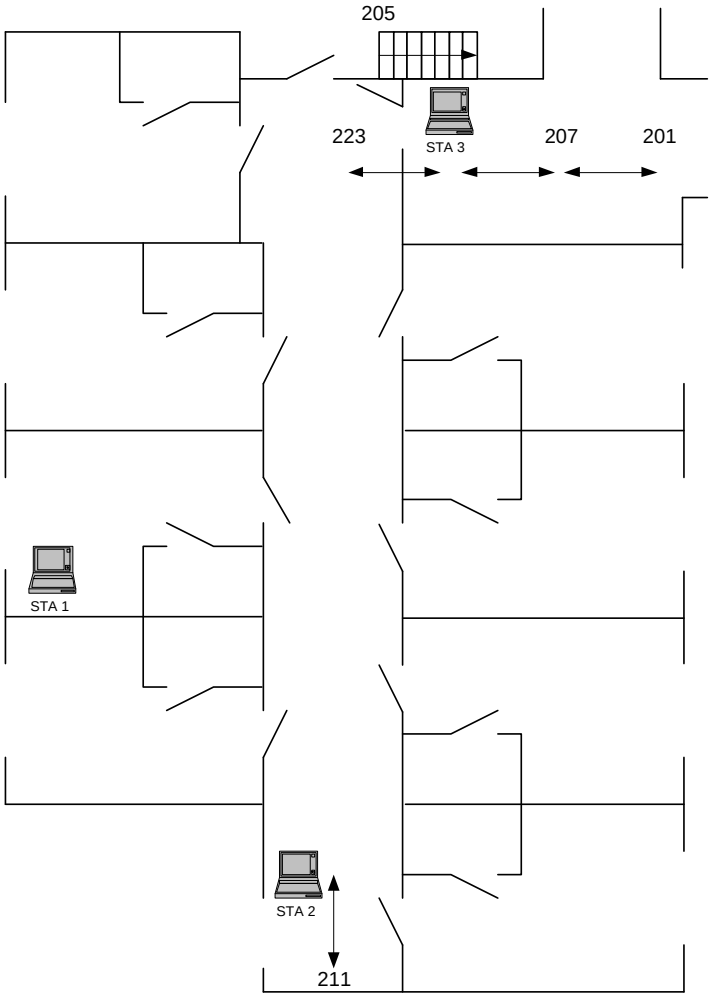

Figure 23: Test area

Table 3 shows logged TTL-values used by the monitor at the Scheduling with quality value test. These values are logged during the time period of 26 seconds. Station STA2 has a larger variation range of TTL-values than STA3, which probably depends on the position the station has with obstacles between it and the AP. Station STA2 and STA3 have both identical image of the Linux kernel and configurations, but there are different hardware (see Table 2). Station STA3 has a minor variation, but showed also a minor area to be moved in, during the preparation

tests. Range values were measured before the: scheduling test with quality values, and is shown around each station. These values show the decrease of signal strength with propagation distance explained in Equation:1. Table 3 gives number of TTL-values each stations received. Variation in these results was probably due to multi path propagation explained in Section 5.

TTL-value	**STA2**	**STA3**
223	17	0
221	5	6
219	13	19
217	7	13
215	3	0
213	1	0

Table 3: Quality value variations during test three.

B Iperf results

Flow	**Test**	**Interv.** (sec)	**Transf.** (MB)	**Bandw.** (Mbps)	**Jitter** (ms)	**Loss** (#)	**Sent** (#)	**Loss** (%)
	1	13,8	4,82	2,94	31,698	5493	8930	62
1	2	13,2	5,53	3,52	18,765	0	3942	0
	3	13,3	3,93	2,48	19,836	6	2812	0,21
	1	13,4	2,80	1,76	15,866	8165	10164	80
2	2	13,7	0,965	0,575	69,383	0	672	0
	3	14,1	0,645	0,373	79,680	0	499	0
	1	13,3	2,23	1,41	12,227	10537	12131	87
3	2	13,2	2,09	1,33	14,937	0	1490	0
	3	13,1	1,74	1,12	14,068	1	1242	0,081
	1	13,0	2,21	1,42	12,444	10587	12163	87
4	2	13,1	2,45	1,57	11,944	0	1749	0
	3	13,0	2,05	1,33	13,434	2	1466	0,14
	1	12,9	2,75	1,78	9,4980	8165	10124	81
5	2	13,1	1,61	1,04	10,320	0	1151	0
	5	12,8	1,43	0,938	9,0100	0	1022	0
	1	12,9	4,60	3,00	4,3060	5676	8954	63
6	2	13,5	1,29	0,799	34,347	0	918	0
	3	12,2	1,13	0,782	37,380	0	809	0

Table 4: Flow specific results for all three tests.

C DSCP field codes

Assured forwarding is the AF is the second service given in Diffserv besides the EF service. The goal with AF is to share the bandwidth capacity into four classes and in each class have three-drop priority levels for packets. DSCP codes explained in Section 4.1.2 for AF at a Diffserv node with PHB is shown in Table 5.

First three bits are for class type of the packet and last three bits are for drop precedence [17].

Importance	Class 1	Class 2	Class 3	Class 4
Low drop	001010	010010	011010	100010
Medium drop	001100	010100	011110	100100
High frop	001110	010110	011110	100110

Table 5: DSCP-codes for AF

D Policing mechanism

Policing is a process of shaping the flow into a defined rate and if this is not possible packets are discarded. Policing of arrival of packets rate in a computer network can be done with the SLB mechanism in TCNG. The SLB uses three parameters to control the rate [12].

- Committed Burst Size (CBS): This is the bucket size of tokens it can be filled with. This is the amount of packets that the flow can send in a peak rate. When this size is reach further tokens are discarded.
- Committed Information Rate (CIR): The rate which tokens are generated into the bucket and also the rate which packets are going to sent into the network.
- Minimum Policed Unit (MPU): This parameter is optional in TCNG and defines the minimum packet size that can be policed.

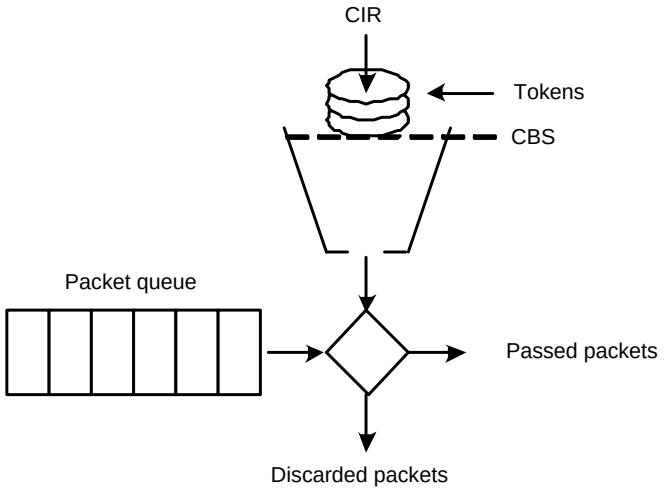

Figure 24: Single Leaky Bucket.

[12] Linux Traffic Control Next Generation, http://tcng.sourceforge.net (Visited 2004-02-23)

E Iptables

A packet way in the Linux kernel with TC and Iptables [13].

[13] Iptables Tutorial 1.1.19, http://iptables-tutorial.frozentux.net (Visited 2004-02-23)

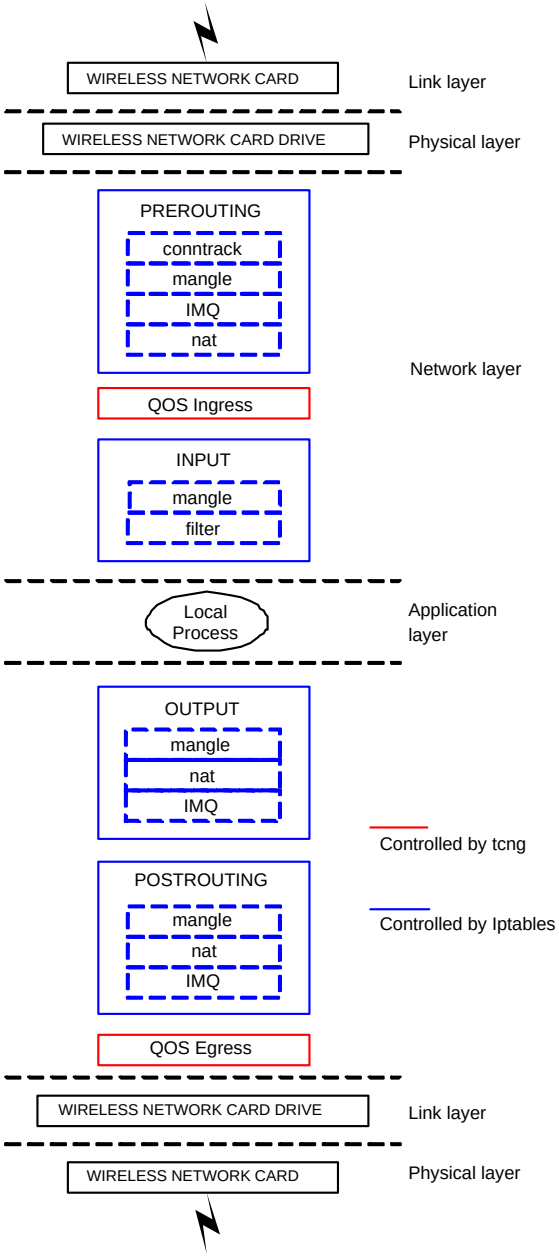

Figure 25: A packets way through Linux kernel with TC and Iptables.