

# Design and Analysis of New Control Mechanisms to Support Data Traffic in Mobile Packet Networks

*György Miklós*

High Speed Networks Laboratory  
Department of Telecommunications and Media Informatics  
Budapest University of Technology and Economics

## **Ph. D. Dissertation Summary**

Advisor:

Dr. Sándor Molnár  
High Speed Networks Laboratory  
Department of Telecommunications and Media Informatics  
Budapest University of Technology and Economics

Budapest, Hungary  
2004

# 1 Introduction

We are experiencing a major paradigm-shift in the telecommunications industry. People are increasingly making use of data applications in addition to the traditional telephony service. A number of popular applications have emerged like email, the World Wide Web and so on, mostly relying on the TCP/IP family of protocols today. Traffic volume of these new data applications is increasing at a much faster rate than traffic volume of telephony. In many places data traffic has already exceeded telephony traffic. It is expected that in the near future data traffic will dominate communication networks in general.

From a technology point of view, this has the consequence that communication devices, architectures and protocols optimized for traditional services like telephony are increasingly being replaced by those optimized for data traffic. In this dissertation we contribute to this shift in technology.

Traffic generated by data applications has basically different characteristics from traffic generated by telephony. Voice calls in traditional systems produce static traffic load for the duration of a call. In contrast, data traffic is dynamic, depending on the nature of the application. Telephony system dimensioning techniques developed by Erlang (see e.g., [1]) do not apply in a data networking context [2]. It has been observed that data traffic exhibits burstiness over many time scales [3].

Thesis 1 is concerned with the dynamic nature of data traffic and proposes a means of characterizing its variability. For this, I consider the *peakedness* measure that was introduced in the context of telephony network dimensioning. I generalize and extend it so that we can use it for data traffic characterization as well. I show how peakedness can be used in discrete time, discuss a number of practical considerations and apply the peakedness measure to a number of traces taken from measurements. In addition, I also provide a technique that gives a Markovian traffic model to fit the peakedness of the measured traffic. I show examples when such a model can capture the peakedness of the traffic over many time scales. I also point out that in some cases a model can be fitted at a given time scale only, indicating long range dependence in the traffic.

The rest of the results are concerned with different design aspects of mobile data networking architectures. The shift towards mobile networking represents another paradigm-shift in the communications industry. Theses 2-4 address some of these design questions within the context of cellular networks. This is the architecture used by first, second and third generation mobile systems as well as Wireless Local Area Networks (WLANs). In the dissertation I consider the architecture of a WLAN, although the presented results could be extended to other private or public mobile networks. We focus on the adaptation mechanisms for wireless links.

Thesis 2 addresses the question of link-layer automatic retransmission (ARQ) protocol. Dynamic traffic patterns and quickly changing error characteristics require that this protocol has to be dynamic as well. I propose a new solution in the HIPERLAN/2 wireless LAN architecture. The proposal is novel due to its dynamic nature: it can adapt the amount of ARQ feedback and the actual content of the ARQ messages based on the traffic and error patterns. Simulation results indicate that the dynamic solution yields higher performance than static solutions.

Thesis 3 analyzes the performance trade-offs in the resource allocation of a wireless base station. Since traffic patterns cannot be predicted, the base station has to make resource decisions in real-time. The wireless channel however makes the problem fundamentally different from that of resource allocation in a fixed network. The amount of service allocated to a user may be different from the amount of service that the user actually gets because of location-dependent channel errors that are typical for an air interface. Channel errors can be

compensated for, but this affects both the total channel utilization and fairness between the users. I introduce a simple compensation mechanism at the base station, and analyze this trade-off, also taking into account the effect of transport layer (TCP) mechanisms and link layer retransmissions.

Thesis 4 extends the analysis of Thesis 3 into a practical scheme. Besides using the compensation for the errors, the properties of the wireless channel are also taken into account to avoid errors. I propose a distributed modular architecture where the wireless channel is monitored by the user terminals that make a decision of their own regarding the use of the channel. The scheduler at the base station is such that it encourages the users to make an efficient use of the resources. I propose a master scheduler and a possible user behaviour algorithm and show by simulation and analysis that the proposed scheme can improve both system utilization and fairness.

Finally Thesis 5 investigates another type of dynamic environment: it considers an ad hoc network where all nodes may be mobile. I propose a new mechanism that enables the use of frequency hopping spread spectrum in an ad hoc networking context. For this, devices may need to dynamically switch between different frequency hopping channels. I present an analytic and simulation-based performance analysis of different channel configurations.

## 2 Research Objectives

Traditional telecommunication networks were often designed to satisfy a well-determined set of application requirements. In the case of modern mobile packet data communication networks however, application needs are hard to predict, and traffic patterns are bursty and unpredictable. This makes it necessary to use dynamic adaptive schemes. The objective of the dissertation is to design and analyze new control mechanisms to support this technological trend.

Specifically, it is the objective of this dissertation to

- investigate methods that characterize the dynamic (bursty) nature of traffic;
- develop solutions that support the efficient transmission of dynamic (bursty) traffic over a wireless channel;
- analyze the trade-off in the resource allocation of a wireless base station with regard to the errors that occur over the air interface;
- design and analyze new resource allocation mechanisms that can maintain control over the trade-off between fairness and utilization, and at the same time support a practical implementation;
- design and analyze new mechanisms that facilitate mobile ad hoc networking using existing radio technology.

## 3 Methodology

The dissertation proposes a number of new characterization methods, control mechanisms and architectures. When investigating the proposals, I have used mathematical analysis as well as simulation-based performance analysis and the analysis of measured traces. I have combined these techniques wherever possible to validate the results and get a deeper insight into the problem.

In Thesis 1 I have used mathematical analysis to develop new traffic characterization and model fitting tools. I have tested the theory on measurement traces taken by other researchers. The test involved computer simulations.

In Thesis 2 I use simulation results to validate the proposed new mechanisms. In Theses 3-4 I have used a combination of analysis and simulations. I have found simulation work to be vital in the analysis. The primary reason for this is that I have investigated the interaction of a number of control loops, including TCP. Due to the complexity of the protocols and the number of adaptation schemes, a complete and accurate mathematical modelling is far beyond the scope of our work. Instead, I have used the power of accurate packet-based simulation for the performance analysis. In addition, to get a deeper insight, I also introduced abstract cases which I analyzed by using mathematical derivations. I have presented how the mathematical models approximate the simulation results. Although the mathematical models neglect many aspects of the proposed scheme, it is nevertheless useful to apply them in approximating the performance of the proposed scheme.

I have followed the same course in Thesis 5. I first give a simple performance model of the contention mechanism supported by simulation, followed by an analytical performance model of some of the possible FHC configurations. The formulae are not expected to completely coincide with the actual protocol performance, but they clearly show the performance trends and explain them. I use packet-level simulations then to validate the protocol and give accurate performance characterization of the protocol. The results are in agreement with the mathematical analysis. I have then used the simulation tool to investigate new cases not covered by the mathematical models.

I have found that the application of mathematical and simulation tools in this way is very constructive and useful in analyzing communication architectures and mechanisms. In practise, the work has consisted of many iterations between the proposed schemes, mathematical modelling and simulations. Each iteration has given new insights and reinforced the strength of this kind of research methodology.

## 4 New Results

### Thesis 1 : Peakedness Characterisation of Bursty Traffic [C1], [C2], [T1], [T2], [D2]<sup>1</sup>, [DA]

An important experience from measurement studies ([3, 4, 2]) regarding the nature of data traffic is that it exhibits bursty properties over many time scales.

One of the key concepts for capturing the bursty character of traffic is self-similarity which resulted in active research on fractal characterization [3, 4]. So far it is not clear how successfully we can utilise self-similarity from a practical traffic engineering point of view but one thing is for sure: burstiness seems to be the most important yet poorly understood characteristic of traffic in high-speed networks. This thesis is motivated by this need. I study peakedness as one of the most promising candidate measures of traffic burstiness.

The simplest burstiness measures take only the first-order properties of the traffic into account. In practice the peak to mean ratio and the squared coefficient of variation are the most frequently used first-order measures [5, C1].

Measures expressing second-order properties of the traffic are more complex. The autocorrelation function, the indices of dispersion [6, 7] and the generalized peakedness [8, 9] are the most well known measures from this class.

---

<sup>1</sup>The notation [Dx] refers to Section x of the dissertation. [DA] refers to Appendix A of the dissertation.

Moreover, there are a number of burstiness measures based on different concepts, e.g. we can use burst length measures [5, 10] or parameters of a leaky bucket for burstiness characterization [11]. By the concept of self-similarity the Hurst parameter and other fractal parameters are also used for burstiness measures [4, 3].

*Peakedness* of a traffic stream has been found a useful characterization tool in blocking approximations and in trunking theory [12]. It has been defined as the variance to mean ratio of the number of busy servers in an infinite hypothetical group of servers to which the traffic is offered, where the service times of the servers are independent and exponentially distributed with a common parameter.

Eckberg [8] extended this definition by allowing arbitrary service time distribution and defined *generalized peakedness* as a functional which maps holding time distributions into peakedness values. For a given complementary holding time distribution  $F^c(x) = P\{\text{holding time} > x\}$ , Eckberg defines the peakedness functional  $z\{F^c\}$  as the variance to mean ratio of the number of busy servers in a hypothetical infinite group of servers with independent holding times distributed according to  $F^c$ . The general definition provides a way to characterize the variability of an arrival stream with respect to a given service system.

**Thesis 1.1 : Generalisation of the Peakedness Measure to Discrete Time [C1], [C2], [T1], [T2], [D2.2.2-2.2.4], [DA]**

I have extended the theory of generalized peakedness to the discrete time domain, and derived the peakedness function for various traffic models.

- First I introduce some notation and definitions.  $w[i]$  is the number of arrivals at epoch  $i$ , where  $i = \dots -1, 0, 1, \dots$ . I assume the stationarity of  $w[i]$ . The first and second moments of  $w[t]$  (independent of  $t$ ) are denoted by  $m_1$  and  $m_2$ . The autocovariance function is  $k[s] = \text{Cov}\{w[i], w[i+s]\} = k[-s]$ .

The service time random variable  $T$  is also discrete and has the distribution  $t[1], t[2], \dots$  on positive integers. (It cannot take on zero value.)  $\mu = 1/E\{T\}$  is the service rate,  $F^c[x]$  is the complementary holding time distribution function and  $\rho_{F^c}[x]$  is the autocorrelation function.

The traffic is offered to an infinite group of servers with independent identically distributed service times determined by  $F^c[x]$ . Each arrival takes a separate server. The peakedness of the arrival stream is defined as the variance to mean ratio of the number of busy servers in the infinite server group:

$$z\{F^c\} = \frac{\text{Var}\{L[t]\}}{E\{L[t]\}} \quad (1)$$

where  $L[t]$  is the number of busy servers at time epoch  $t$ .

An important modification of the definition is to let the service time depend on the arrival epoch only (have a common service time for all  $w[t]$  arrivals at epoch  $t$ ). I call (in accordance with [13]) the peakedness value defined in this way the *modified* peakedness  $\tilde{z}\{F^c\}$ .

- I have shown [DA] that

$$\tilde{z}\{F^c\} - z\{F^c\} = \left(\frac{m_2}{m_1} - 1\right) (1 - \rho_{F^c}[0]\mu). \quad (2)$$

that is, their difference is constant (cf. (35) in [13]). The importance of this modified definition lies in the fact that it gives a way to handle a whole batch of arrivals together,

which can save a lot of computational effort in the case of measuring the peakedness for a general holding time distribution. Below, I will keep the original definition of peakedness (eq. (1)).

- The most important case in discrete time is the case of geometrically distributed holding times:  $t[i] = \mu(1 - \mu)^{i-1}$ ,  $0 < \mu < 1$ .

Let us introduce the notation

$$K[s] = \begin{cases} \frac{2}{m_1}k[s] & \text{if } s > 0 \\ \frac{1}{m_1}k[0] & \text{if } s = 0 \end{cases}$$

and let its z-transform be  $K^*(\omega) = \sum_{s=0}^{\infty} K[s]\omega^s$ .

- I have derived [DA] that the peakedness function of the arrival stream with respect to geometric holding time distribution is given by

$$z_{\text{geo}}(\mu) = 1 + \frac{K^*(1 - \mu) - 1}{2 - \mu} \quad (3)$$

- I have shown that there exists a connection between the index of dispersion for counts (IDC) and the peakedness of an arrival stream with geometric holding time distribution. The IDC is a measure used to characterize the variability of an arrival stream on different time scales, and is defined as

$$I[t] = \frac{V[t]}{E[t]} = \frac{V[t]}{m_1 t} \quad (4)$$

where  $E[t]$  and  $V[t]$  are the mean and variance of the number of arrivals in  $t$  consecutive epochs ( $t = 1, 2, \dots$ ).

I have derived that the connection between the peakedness function and the IDC is [DA]

$$z_{\text{geo}}(\mu) = 1 - \frac{\mu^2 \frac{dI^*(1-\mu)}{d\mu} + 1}{2 - \mu} \quad (5)$$

where  $I^*(\omega)$  is the z-transform of  $I[t]$ .

I have derived asymptotic results based on eq. (5). If  $\lim_{s \rightarrow \infty} I[s]$  exists and is non-zero, then I have:

$$z_{\text{geo}}(0) = \frac{\lim_{s \rightarrow \infty} I[s] + 1}{2} \quad (6)$$

$$z_{\text{geo}}(1) = I[1] = \frac{\text{Var}\{w[i]\}}{\text{E}\{w[i]\}}. \quad (7)$$

- I have determined the peakedness values for a number of traffic models [DA],[C2]. Specifically, for the Markov Modulated Batch Bernoulli Process (MMBBP), the peakedness is computed as follows. Let  $\mathbf{P}$  and  $\mathbf{D}$  denote the transition probability matrix and the steady-state distribution vector of the modulating Markov process, respectively ( $\mathbf{D}\mathbf{P}=\mathbf{D}$ ). Let  $\mathbf{M}_1$  and  $\mathbf{M}_2$  be diagonal matrices corresponding to the first and second moments of the number of arrivals in the corresponding states. Let  $\mathbf{e}$  be a vector of all ones and let  $\mathbf{I}$  be the identity matrix.

We can express the mean number of arrivals as  $m_1 = \mathbf{D}\mathbf{M}_1\mathbf{e}$  and the second moment as  $m_2 = \mathbf{D}\mathbf{M}_2\mathbf{e}$ . The autocovariance function of the arrival process is given by  $k(i) = \mathbf{D}\mathbf{M}_1\mathbf{P}^i\mathbf{M}_1\mathbf{e} - m_1^2$ .

- I have derived [DA] that the peakedness function is

$$z_{\text{geo}}(\mu) = 1 + \frac{1}{2 - \mu} \left( \frac{2(1 - \mu)\mathbf{D}\mathbf{M}_1\mathbf{P}(\mathbf{I} - (1 - \mu)\mathbf{P})^{-1}\mathbf{M}_1\mathbf{e} + m_2}{m_1} - 1 \right) - \frac{m_1}{\mu} \quad (8)$$

- I have also derived the peakedness of some important special cases: the Batch Bernoulli Process (BBP), the Markov Modulated Bernoulli Process (MMBP) and the Switched Batched Bernoulli Process (SBBP) [D2.2.4].
- Another important traffic model is the batch renewal process. It is important to consider because of its ability to model the correlation structure of traffic [14]. The discrete time batch renewal process is made up of batches of arrivals, where the intervals between batches are independent and identically distributed random numbers, and the batch sizes are also independent and identically distributed, furthermore, the batch sizes are independent from the intervals between batches.

Let us use the following notation for the discrete time batch renewal process:  $a$  and  $b$  are the mean length of intervals between batches and the mean batch size, respectively. The first and second moments of the number of arrivals in an epoch is given by  $m_1 = b/a$ , and  $m_2 = m_1 b(C_b^2 + 1)$  where  $C_b^2$  is the squared coefficient of variation (variance to mean square ratio) of the batch size. The probability generating function of the distribution of time between batches is denoted by  $A^*(\omega)$ . I have derived [DA] the peakedness for geometric holding times which is given by

$$z_{\text{geo}}(\mu) = 1 + \frac{1}{2 - \mu} \left( \frac{1 + A^*(1 - \mu)}{1 - A^*(1 - \mu)} - b + \frac{m_2}{m_1} - 1 \right) - \frac{m_1}{\mu} \quad (9)$$

**Thesis 1.2 : New Model Fitting Technique Based on the Peakedness Measure.**  
[C2], [T1], [T2], [D2.2.5-2.3]

I have developed a new model fitting technique based on the peakedness measure of traffic, and successfully applied it to measured traffic traces.

The technique is based on the mean rate  $m_1$  of the arrival traffic, the peakedness value at  $\mu = 1$  and at three other points,  $\mu_1, \mu_2, \mu_3$ . The model that I fit to the peakedness curve is an Interrupted Batch Bernoulli Process (IBBP): in one state of the modulating Markov process, the arrival number has a general distribution, in the other state, there are no arrivals.

- First, by  $z(1) = m_2/m_1 - m_1$ , we get  $m_2$ . Introducing  $\omega = 1 - \mu$ ,  $\omega_i = 1 - \mu_i$  we can compute (using the values  $K^*(\omega_i) = (z_{\text{geo}}(\mu_i) - 1)(\omega_i + 1) + 1$ )

$$Y_i = Y(\omega_i) = m_1 \frac{1 - \omega_i}{2\omega_i} \left( K^*(\omega_i) + m_1 \frac{1 + \omega_i}{1 - \omega_i} - \frac{m_2}{m_1} \right) \quad (10)$$

By applying eq. (8) to the IBBP case, we get,

$$Y(\omega) = m_* - \frac{(m_* - m_1^2)(1 - \gamma)}{1 - \gamma\omega}. \quad (11)$$

Let us denote

$$\tilde{Y} = \frac{Y_1 - Y_2}{Y_2 - Y_3} \quad (12)$$

which evaluates to

$$\tilde{Y} = \left( \frac{\omega_2 - \omega_1}{\omega_3 - \omega_2} \right) \left( \frac{1 - \gamma\omega_3}{1 - \gamma\omega_1} \right) \quad (13)$$

and we get

$$\gamma = \frac{\tilde{Y} \frac{\omega_3 - \omega_2}{\omega_2 - \omega_1} - 1}{\tilde{Y} \frac{\omega_3 - \omega_2}{\omega_2 - \omega_1} \omega_1 - \omega_3} \quad (14)$$

Once we have  $\gamma$ , we can obtain an estimation for  $m_*$  as

$$m_* = \frac{1}{3} \sum_{i=1}^3 \frac{Y_i - \frac{m_i^2(1-\gamma)}{1-\gamma\omega_i}}{1 - \frac{1-\gamma}{1-\gamma\omega_i}} \quad (15)$$

where we have on the right hand side an average for the known values  $\omega_i, Y_i$ .

- I propose to fit an IBBP (no arrivals in state 2) as follows:

$$m_{1,(1)} = \frac{m_*}{m_1}, \alpha_2 = \frac{m_1(1-\gamma)}{m_{1,(1)}}, \alpha_1 = 1 - \gamma - \alpha_2, m_{1,(2)} = m_2 \frac{\alpha_1 + \alpha_2}{\alpha_2}. \quad (16)$$

Given the first and second moments of the number of arrivals in state 1, we can use for example a generalized geometric distribution for modelling the batch size distribution. In this case, there are no arrivals with probability  $1 - \varphi$ , and there is a batch of arrivals with geometrically distributed size of parameter  $\psi$ . The moments are given by

$$m_{1,(1)} = \varphi/\psi, \quad (17)$$

$$m_{1,(2)} = \varphi/\psi^2 \quad (18)$$

by which we can get  $\varphi, \psi$  for the model.

If it is possible to exactly fit an IBBP to the  $\mu_i, z_{\text{geo}}(\mu_i)$  pairs, the values that are summed in the equation for  $m_*$  are identical. If there is no IBBP that exactly fits the given peakedness values,  $m_*$  gives an estimation and the peakedness curve of the fitted IBBP model approximates the  $\mu_i, z_{\text{geo}}(\mu_i)$  pairs.

- I have made a number of empirical peakedness measurements on data traffic traces using a simulation of the hypothetical service. I have shown that the peakedness curve (using geometric service distribution and varying the service rate) shows the variability of the traffic on different time scales. This is clearly visible in the peakedness curves of MPEG video traces, where the variability of the frame sequence and Group of Pictures (GOP) sequence can be differentiated, and the impact of the type of movie on the peakedness curve is seen. I have successfully applied the model fitting technique to the video traces to capture its peakedness curve.
- I have applied the model fitting technique to data traffic traces as well. In the case of ATM and Ethernet traces, I have shown that a Markovian model can be fitted to the peakedness curve only at a given time-scale. This indicates that the peakedness curve can be used for detecting long range dependence in the traffic.

## Thesis 2 : Design of a New Dynamic Retransmission Protocol [C5], [S1]-[S6], [P1], [P2], [P3], [D3]

In the HIPERLAN/2 wireless LAN system the ARQ protocol has to provide feedback to a dynamically changing amount of traffic in such a way that it is sent in control PDUs that are scheduled separately from data traffic. Therefore, new mechanisms are needed to optimize the contents of the ARQ messages and schedule them. These mechanisms must work even if there are implementation-dependent delays in the end systems.

To satisfy these new demands, a new version of selective repeat ARQ has been defined, called Selective Repeat ARQ with Partial Bitmaps (SRPB) [C5, 15]. The protocol allows the transmission and reception of PDUs in a very flexible way within the transmit and receive windows, respectively.

- I have proposed a flexible selective feedback format for ARQ messages. The receiver maintains a bitmap corresponding to the reception status of each PDU in its receive window and uses this bitmap to give feedback to the transmitter. Based on this feedback, the transmitter retransmits the missing PDUs. The ARQ feedback in ARQ C-PDUs with the information fields are summarized in Table 1. In a single ARQ C-PDU, the receiver can signal three 8-bit portions of this bitmap, called bitmap blocks. Each bitmap block is identified by a bitmap number. In addition, the receiver can cumulatively acknowledge earlier PDU receptions by the CAI (Cumulative Ack Indication) bit. When it is set, all PDUs before the first bitmap block have been correctly received.

Field	Bits	Function
CAI	1	Cumulative Ack Indication
BMN 1	7	Bitmap Number 1
BMB 1	8	Bitmap Block 1
BMN 2	5	Bitmap Number 2
BMB 2	8	Bitmap Block 2
BMN 3	5	Bitmap Number 3
BMB 3	8	Bitmap Block 3
ABIR	1	ARQ Bandwidth Increase Request

Table 1: ARQ feedback fields

- I have proposed three simple schemes that the receiver can follow to select the bitmap blocks to signal in the ARQ message.
  - A In the first and simplest strategy, the receiver always signals the blocks continuously from the bottom of the window. This is the simplest strategy to follow, but it does not utilize the flexibility of the ARQ messages. This strategy is illustrated in Figure 1.
  - B In the second strategy, the receiver signals the blocks beginning from the bottom of the window, but only those where there is an error. This makes the retransmission of errored PDUs faster. This strategy is illustrated in Figure 2.
  - C The third strategy is different for the case when the round-trip time of the ARQ protocol is greater than one frame. The round-trip time is the minimum time

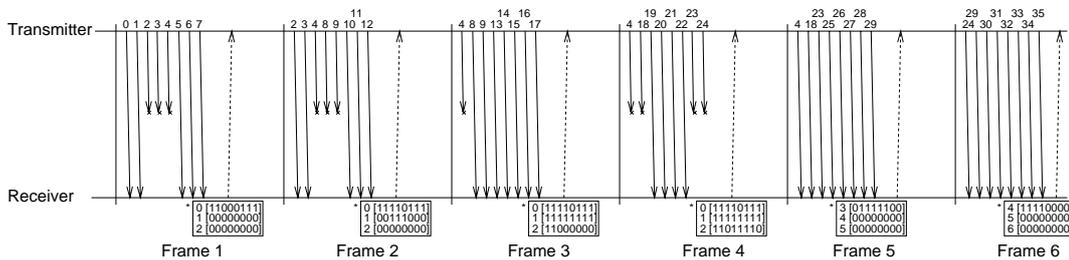


Figure 1: Receiver ARQ strategy A. The receiver chooses the first three bitmap blocks to signal in the ARQ control message. (The bitmap blocks are shown in brackets, preceded by the bitmap block numbers.)

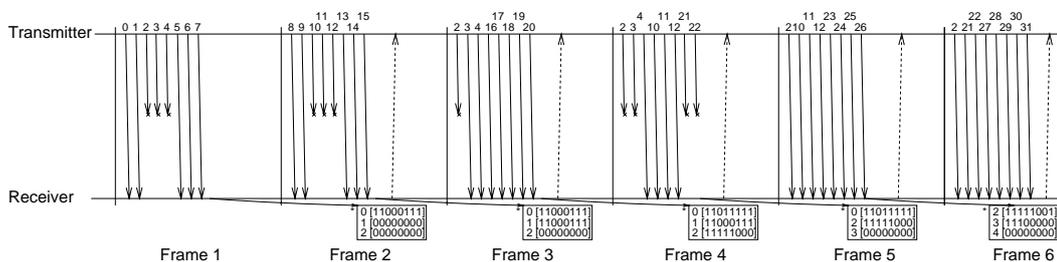


Figure 2: Receiver ARQ strategy B. The receiver chooses the first three bitmap blocks where there is a missing PDU. The receiver has a delay of one frame for generating the ARQ C-PDU.

necessary for a retransmission to arrive at the receiver after the negative acknowledgement is sent. The round-trip time of the ARQ protocol can be bigger than one frame because of processing delays in the transmitter or receiver. In this strategy, the receiver signals the blocks from the bottom of the window, but only those where there is an error, and excluding those which been signalled in the last round-trip time period (including this frame) and are unchanged. In addition, the receiver chooses the first bitmap block in the window at least once in a round-trip time to be able to cumulatively acknowledge all data below the window.

The motivation for this rule is to avoid signalling the same blocks twice within a round-trip time, since the retransmission can not arrive earlier. Instead, I utilize the ARQ signalling capacity for the signalling of new feedback information. This strategy is illustrated in Figure 3.

- We have implemented strategy B in a packet-based simulator and compared it to PRIME ARQ in [C5]. The results show the throughput performance of the SRPB and PRIME ARQ using the same set of constraints: a single cell was simulated on a bursty Markovian error model. As the offered traffic is increased, the saturation throughput is at a higher value for the SRPB protocol by approximately 20%.
- I have proposed a scheme that allows the dynamic setting of the ARQ signalling capacity at the base station using a one-bit feedback from the receiver. Allocating more ARQ C-PDUs allows quicker retransmissions, but it also consumes more resources. This dynamic allocation of ARQ capacity is supported by the ABIR (ARQ Bandwidth Increase Request) bit in the ARQ C-PDU. When this bit is set in an ARQ message

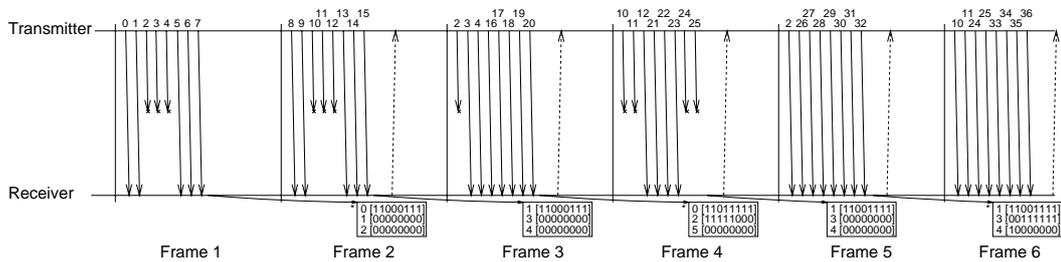


Figure 3: Receiver ARQ strategy C. The receiver excludes the bitmap blocks that have been signalled in the last round-trip time and are unchanged. The acknowledgements where the CAI (Cumulative Ack Indicator) is set are marked by an asterisk. At least one such ARQ C-PCU is generated in each round-trip time.

sent by an MT, it signals to the AP scheduler that the MT would like to increase the number of ARQ C-PDUs.

In the adaptation mechanism, when a receiver can not report all the missing PDUs in the ARQ feedback message of a single frame, it signals this to the scheduler in the AP by setting the ABIR bit. The number of allocated ARQ C-PDUs is increased by one when the ABIR bit is set, otherwise it is reduced, but at least one ARQ C-PDU is always allocated. This allows a quick adaptation of the number of ARQ C-PDUs, which is more efficient than a static allocation. Simulation results have verified that the scheme improves the throughput of the system, by up to 100% depending on the error pattern.

### Thesis 3 : Fairness and Utilization Analysis of the Resource Allocation of a Wireless Base Station [C3], [C4], [T3], [D4]

The question of fair resource allocation has been extensively studied in wired networks using the notion of fluid fair queuing. Using this concept, an ideally fair resource allocation can be defined among the flows sharing a common link [16, 17].

As we apply the fair queuing algorithms and results in a wireless environment, new problems arise. Because packets may be lost at the air interface, the amount of service allocated to a flow is not the same as the amount of service that the flow eventually gets. It implies that the fairness of fluid fair queuing no longer holds. Since the channel error characteristics may be location-dependent, the service that the flows get are not proportional to their weights.

We investigate this question in the resource allocation architecture illustrated in Figure 4. This architecture is in harmony with the HIPERLAN/2 wireless LAN system [18, 19, 20, 21] [D3.2], but the resource allocation architecture itself is general and can be applied in other systems as well.

I call each transmission/reception entity a user. Each connection of a MT (Mobile Terminal) corresponds to a user. The base station (BS, which is an Access Point, AP in the HIPERLAN/2 architecture) is represented by many users each belonging to a connection interfacing a MT. The BS contains the master scheduler that is responsible for the distribution of the capacity between the users. The output of the scheduler is the dynamic allocation of each fixed-length MAC frame. The unit of allocation is one radio PDU which is of fixed length.

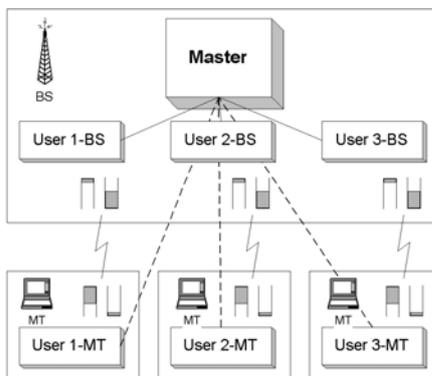


Figure 4: Resource allocation architecture

The master scheduler makes its resource allocation decisions based on the resource requests from the users. For the users within the BS the resource requests are just internal information within the base station. For the users in the MTs the resource requests are made to the master scheduler as control information. Resource requests specify the amount of data that the user has in its buffers to transmit.

The master scheduler runs the scheduling algorithm once for each frame. This algorithm takes the resource request from the users as its input and determines how much capacity is allocated to each user in the next frame in the unit of a radio PDU. Here I do not deal with the order of allocations within a frame. Our focus is the amount of allocations determined by the master scheduler for each frame.

- I have proposed a new simple compensation mechanism for the analysis of the trade-off between fairness and utilization in the resource allocation of a wireless base station. The master scheduler gives additional allocations to the users which have experienced errors. Errors become known to the master scheduler through the ARQ protocol after the errors occur.
- I have implemented the compensation mechanism by extending SFQ (Start-time Fair queuing). I introduce the state variable *lag* for each user, denoted by  $n_i$ , to represent the amount of normalized service that the user should get in compensation. The constant  $\beta_i$ ,  $\beta_i \leq 1$  represents the amount of compensation for lost resources.  $\beta_i = 1$  means that all of the lost capacity is compensated later,  $\beta_i = 0$  means no compensation for lost capacity. It is even possible to use negative compensation,  $\beta_i < 0$ .
- The proposed compensation mechanism runs as follows [D4.4-4.5]:
  - 1 A start and a finish tag,  $S_i$  and  $F_i$ , are associated with each *user*, corresponding to the virtual start and finish time of the packet at the head of the queue. When a new packet enters the head of the queue at time  $t$  (i.e., a packet has been served, or the user becomes backlogged), then the new values are computed from the old value of the finish tag,  $F_i'$ , as

$$S_i \leftarrow \max\{F_i', v(t)\} \quad (19)$$

where initially  $F_i' = 0$ ,  $w_i$  is the weight of flow  $i$ ,  $L$  is the length of all packets, and  $v(t)$  is the virtual time at time  $t$ .

- 2 The server virtual time is initially 0. During a busy period the server virtual time at time  $t$ ,  $v(t)$ , is defined to be equal to the start tag of the packet in service at time  $t$ . At the end of a busy period,  $v(t)$  is set to the maximum of finish tag assigned to any packets that have been serviced by time  $t$ .
- 3 Packets are served in increasing order of start tags; ties are broken arbitrarily.
- 4 After each error of a packet of length  $L$  on the wireless channel for user  $i$  as reported by the ARQ protocol, its lag is incremented by  $\beta_l$  times the normalized service that was lost:

$$n_i \leftarrow n_i + \beta_l \frac{L}{w_i} \quad (20)$$

$$n_i \leftarrow \min\{n_i, n_{max}\}, \quad n_i \leftarrow \max\{n_i, -n_{max}\}. \quad (21)$$

- 5 The finish tag is computed as follows:

$$l_c \leftarrow \min\{n_i, \beta_l \frac{L}{w_i}\}, \quad l_c \leftarrow \max\{l_c, -\beta_l \frac{L}{w_i}\}, \quad (22)$$

$$F_i \leftarrow S_i + \frac{L}{w_i} - l_c, \quad n_i \leftarrow n_i - l_c, \quad (23)$$

where  $l_c$  is the normalized compensation given during the service of the packet. Its value is  $\beta_l \frac{L}{w_i}$  when  $n_i$  is positive,  $-\beta_l \frac{L}{w_i}$  when  $n_i$  is negative.

- I have also shown that the admission criterion for a new flow (based on the worst case scenario when all flows need compensation) evaluates to

$$\frac{1}{1 - \beta_l} \sum_{i \in \mathcal{U}} w_i \leq C. \quad (24)$$

- Using the compensation mechanism proposed in above, I have performed the simulation based performance analysis of the trade-off between fairness and utilization at the wireless base station. In the simulation study, I have used a number of flows with different weights, each carrying a greedy TCP session. I have implemented a link-layer ARQ scheme, a simplified version of the SRPB protocol proposed in Thesis 2. I have investigated the system utilization and fairness with respect to the allocated throughput and throughput at the DLC, IP and application levels. The results have verified that the proposed compensation mechanism indeed controls the trade-off between fairness and utilization. From the simulation results, I have concluded the following.
  - I have found that fairness of application level throughput values can be considerably improved by positive compensation at the expense of reduced system utilization, even by an order of magnitude using the parameters and metrics defined in [D4.6].
  - The reduction of utilization is slight in the case of independent channel losses (1.8%), and increases with the burstiness of the loss model (10.3%).
  - Higher utilization (by 3.8%) is achieved by applying negative compensation or by using a semi-reliable ARQ at the link layer.
  - The fairness of allocation can become considerably worse and can not be improved by the compensation mechanism in case of using semi-reliable ARQ.

## Thesis 4 : Distributed Scheme for the Resource Allocation of a Wireless Base Station [C6], [C7], [D5]

### Thesis 4.1 : New Distributed Resource Allocation Architecture [C6], [D5.1-5.6]

I have proposed a novel resource allocation scheme that follows a distributed approach. The master scheduler in the base station does not have any information about the current state of the wireless channel. Instead it compensates for channel errors after they occur. I extend the architecture by *allowing the users to defer their transmission to a later time when the channel is temporarily in a bad state*. The approach is therefore decentralized in the sense that the master scheduler using a simple compensation mechanism in making the scheduling decision without any regard to the wireless channel state. Each user of the channel is responsible by itself for the estimation and prediction of its own channel, and can optimize when to transmit on its own. When the channel is expected to be bad for some time, the user can defer its transmission until the channel is expected to recover, based on the measured channel properties. This is encouraged (but not controlled) by the master scheduler as it allows the users to partially reclaim unused capacity in the future.

This approach offers several key advantages.

- We do not need to make any assumptions about the error characteristics of the channel and the master scheduler does not need the prediction of the state of the link.
- Our scheme offers a modular implementation and greatly simplifies the master scheduler.
- It offers a decoupling of functionality: the users' estimation and prediction of the channel can be changed without modifying the master scheduler.

I have given guidelines on how to apply a wireline fair scheduling algorithms in the master scheduler of a wireless base station. Based on these guidelines, I have proposed the application and extension of SFQ, Start-time Fair queuing. This extension includes, besides the compensation for lost resources as in Subthesis 3.1, another type of compensation: compensation for unused resources.

In the master-scheduler, I keep constant  $\beta_l$  of Thesis 3 to represent the amount of compensation for lost resources. Recall that  $\beta_l = 1$  means that all of the lost capacity is compensated later,  $\beta_l = 0$  means no compensation for lost capacity (we now use values,  $0 \leq \beta_l \leq 1$ ). I introduce a new constant  $\beta_u$  to represent the amount of compensation for unused resources,  $0 \leq \beta_u \leq 1$ . Besides the constants  $\beta_l$  and  $\beta_u$ , I also study the effects of the speed of compensation represented by  $\gamma$ ,  $0 \leq \gamma \leq 1$ , which determines the increase of allocations when a user is being compensated.  $\gamma = 0$  corresponds to no compensation, whereas  $\gamma = 1$  corresponds to immediate compensation, where a user is compensated before any other users can get more allocations. Each user can observe the quality of its own channel through measurements and can decide how much capacity to request.

The proposed master-scheduler is a modification of the scheduler of Subthesis 3.1 as follows.

- 1 A start and a finish tag,  $S_i$  and  $F_i$ , are associated with each *user*, corresponding to the virtual start and finish time of the packet at the head of the queue. When a new packet enters the head of the queue at time  $t$  (i.e., a packet has been served, or the user becomes backlogged), then the new values are computed from the old value of the finish tag,  $F'_i$ , as

$$S_i \leftarrow \max\{F'_i, v(t)\} \quad (25)$$

where initially  $F'_i = 0$ ,  $w_i$  is the weight of flow  $i$ ,  $L$  is the length of all packets, and  $v(t)$  is the virtual time at time  $t$ .

2 The server virtual time is initially 0. During a busy period the server virtual time at time  $t$ ,  $v(t)$ , is defined to be equal to the start tag of the packet in service at time  $t$ . At the end of a busy period,  $v(t)$  is set to the maximum of finish tag assigned to any packets that have been serviced by time  $t$ .

3 Packets are served in increasing order of start tags; ties are broken arbitrarily.

4 After each error of a packet of length  $L$  on the wireless channel for user  $i$  as reported by the ARQ protocol, its lag is incremented by  $\beta_l$  times the normalized service that was lost:

$$n_i \leftarrow \min\left\{n_i + \beta_l \frac{L}{w_i}, n_{max}\right\}. \quad (26)$$

5 When user  $i$  becomes unsatisfied at the beginning of a frame at time  $t$  after being satisfied, its lag is incremented by  $\beta_u$  times the normalized service that the user missed:

$$n_i \leftarrow \min\left\{n_i + \beta_u \max\{v(t) - F'_i, 0\}, n_{max}\right\} \quad (27)$$

where  $v(t)$  is the virtual time at time  $t$ , and  $F'_i$  is the finish time of user  $i$  before it became satisfied. Since  $v(t)$  can be interpreted as the normalized fair amount of service that each user could have received up to time  $t$ ,  $v(t) - F'_i$  is the amount of normalized service that the user missed while it was satisfied.

6 The finish tag is computed as follows as follows.

$$l_c \leftarrow \min\left\{n_i, \gamma \frac{L}{w_i}\right\}, \quad (28)$$

$$n_i \leftarrow n_i - l_c, \quad (29)$$

$$F_i \leftarrow S_i + \frac{L}{w_i} - l_c, \quad (30)$$

where  $l_c$  is the normalized compensation given during the service of the packet.

I have proposed a user behaviour that complements the scheduler in the base station. A user either makes a resource request according to the packets waiting for transmission in its buffers, or makes a resource request of zero, thereby relinquishing service to a later frame. The proposed solution consists of a method for a user to decide when to relinquish service (i.e., make a resource request of 0). We assume that a user has knowledge of the scheduling algorithm and its constant parameters.

Relinquishing service in a given frame can be useful for the mobile because channel behaviour is typically positively correlated, so when a user observes bad channel, it is likely that the channel will continue to be bad for some time.

In the proposed solution [D5.5], the user builds a model of the channel estimating the parameters, which enables it to make a prediction of the future expected channel state and decide when to relinquish service. The model is an AR(1) model that captures the correlation properties of the channel as well. The proposed solution uses thresholds on the measured success rate and decides to relinquish service when the expected compensation in the future is likely to improve performance compared to the immediate transmission of user data.

I have also proposed a one-bit feedback mechanism [D5.5.3] from the master scheduler to the user that facilitates the adaptive setting of the threshold. The advantage of this adaptive sensitivity setting is that the users can maximize their throughput by setting their sensitivity to the level where they request just as much compensation as possible.

I have made a simulation based performance analysis of the proposed architecture. I have taken simulation traces that verify the user behaviour algorithm. I have made a number of simulations that show the increase of system throughput as a result of the compensation mechanism in the range of 10%-20%. The simulation results show the performance dependence on the parameter settings.

- I have shown that the throughput performance of a user has a maximum as a function of the sensitivity of the user behaviour algorithm. If the user relinquishes its requests too infrequently, then the user is not efficient in avoiding bad channel states. If the user relinquishes its requests too often, then the compensation mechanism gets saturated and the user can not get back the compensation. This follows that there is an optimal sensitivity for the user behaviour algorithm. (In one example of parameter settings, this happens when the relinquishing frequency is 0.25). I call this optimal point the locally greedy user behaviour since it optimizes the throughput of a single user from a greedy point of view.
- I have shown that the effect of changing the speed of compensation, using the adaptive sensitivity setting, is that there is time for more frequent relinquishing of service (its rate goes from 0.05 to 0.35 with our particular set of parameters).
- I have shown that the proposed architecture can improve the throughput performance of users with both bursty and independent error characteristics. (In our simulations, we achieved improvements in the range of 10%-20%). In the case of bursty channel errors, the user behaviour could make use of prediction to avoid bad states. In the case of independent errors, the improvement is from the other users making better use of the available capacity and because of the compensation for lost resources.

#### **Thesis 4.2 : Analytical Approximation of System Performance [C6], [C7], [D5.7]**

I have proposed an approximation that is analytically tractable and provides an approximation to the system performance.

- I have defined two abstract user behaviour cases, the “no algorithm” and “ideal algorithm” cases. The “no algorithm” case is a simple one when the user does not relinquish service at all. The ideal algorithm is one in which with no effective relinquishing of service, the success rate is no smaller than what can be achieved by any rate of relinquishing service. Such an algorithm is naturally impossible to implement or even approach, but it provides a valuable abstraction for the analysis of the system.
- I have analytically derived the allocated and application level throughput values for the users as well as the system utilization and fairness measures as follows.

Let us denote by  $\Pi_i$  the transmission success rate of user  $i$ . The compensation can be modelled by the following modified weight of user  $i$  [C7]:

$$w'_i = \frac{1}{1 - \min\{\gamma, \beta_l(1 - \Pi_i)\}} w_i. \quad (31)$$

The total system capacity,  $C$ , is distributed among the users according to equation 31. Using  $W' = \sum_i w'_i$ , the allocation and goodput to user  $i$  (assuming greedy users) are

$$a_i = \frac{w'_i}{W'}C \text{ and } g_i = \Pi_i a_i. \quad (32)$$

I use the following metrics to characterize the total system performance: the system utilization, defined by  $U = \sum_i g_i/C$ , the goodput fairness  $V_g = \sigma\{g_i/w_i\}/E\{g_i/w_i\}$ , and allocation fairness  $V_a = \sigma\{a_i/w_i\}/E\{a_i/w_i\}$ , that is, the coefficient of variation of normalized goodput and allocation values.

For a specific channel, we can use the following success ratios:  $\Pi_{i,avg}$  for the average success rate of the channel, and  $\Pi_{i,max}$  for the maximum success rate that can be achieved by any user behaviour. Such an upper limit can be interpreted, for example, for a Markovian channel by the success rate in the best state. By using the success rates  $\Pi_{i,avg}$  and  $\Pi_{i,max}$  for  $\Pi_i$ , the utilization and fairness metrics can be derived for the no algorithm and ideal algorithm cases.

- I have compared the results with the simulated ones and I have shown that the abstract analysis approximates the system performance. This validates the use of the analysis above.

Using these system performance metrics, I have shown that the proposed architecture can improve system utilization at the same time as improving the fairness of the system. This is possible because the proposed architecture encourages the efficient utilization of system resources.

## **Thesis 5 : A Novel Scheme to Interconnect Multiple Frequency Hopping Channels into an Ad Hoc Network [J2], [P6], [P7], [D6]**

In the theses above, I have investigated packet-switched networks that all possessed a fixed infrastructure. Using wireless technologies, however, it is possible to design networks where such a pre-installed fixed infrastructure is not present. These type of networks are referred to as ad hoc networks. The MANET (Mobile Ad hoc Networking) working group of the IETF has been formed to study the protocol issues involved in such networks. The vision of the MANET working group [22] “is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multihop topologies which are likely composed of relatively bandwidth-constrained wireless links.”

### **Thesis 5.1 : Multiple Frequency Hopping Channel Communication [J2], [D6.3]**

I have proposed Multiple Frequency Hopping Channel communication (MFHC), a scheme that forms a connected ad hoc network from multiple frequency hopping channels. The scheme relies on the notion of home FHC. Each device participating in an ad hoc network has a home FHC which determines the frequency hopping scheme it follows whenever it is not transmitting at another FHC. To transmit to a particular device, it is necessary to switch to that particular device’s home FHC, listen to the channel and resolve contention based on an adapted CSMA/CA scheme. Compared to earlier systems, this architecture has the advantage of using low cost frequency hopping radios as in Bluetooth [23], based on a simple connection-less approach with on-demand resource allocation scheme as in the case of IEEE 802.11 [24], which enables networking between all devices as in Hop-Reservation

Multiple Access [25] and High Frequency Intra Task Force Communication Network [26], but without the need for a network-wide synchronization mechanism. None of the earlier systems possesses all these properties.

Channel access within a FHC is based on the CSMA/CA approach used by the IEEE 802.11 protocol [24]. This means that a node that has a packet to send on the FHC first waits until the channel becomes free for at least a minimum period of time, referred to as GS (guard space). Communication may begin at fixed slot boundaries. (I do not specify the length of a slot here, and simply use slots as the unit time on a FHC.) To resolve collisions due to more than one stations sending at the same time, a contention mechanism is applied as follows. Each station has a contention window,  $CW$ , and chooses a random backoff value  $B$  from the interval  $[0, CW - 1]$ . In each slot when the channel is sensed free, the value of  $B$  is decreased if it is above zero. A node may transmit when the value of  $B$  reaches zero. If the transmission is successful, the value of  $CW$  is initialized to  $CW_{min}$ . If the transmission is unsuccessful, the value of  $CW$  is doubled unless it reaches  $CW_{max}$ . This scheme ensures that collisions will be resolved after one or more stages of contention.

We precede each packet transmission by an RTS-CTS message exchange, as in the 802.11 protocol. This handles the hidden terminal problem (the destination receives packets from a station that the source cannot receive from), and also decreases the overhead of contention in the case of long packets. In addition, the RTS-CTS message exchange provides a way for negotiation of parameters for the subsequent data transmission.

I have extended this scheme for multiple FHCs, as shown in the example of Figure 5. Even though it is allowed for a node to switch from one FHC to another, a home FHC is associated with each node. The figure shows two FHCs, where FHC 1 is the home of nodes A and B, FHC 2 is the home of nodes C, D and E. A node may temporarily leave its home FHC, as node B does to visit FHC 2 (B'), but it returns to its home FHC as soon as it has finished contention or transmission. To initiate a data transmission to a node, we need to switch to the destination node's home FHC and wait until the node is available and the channel is free.

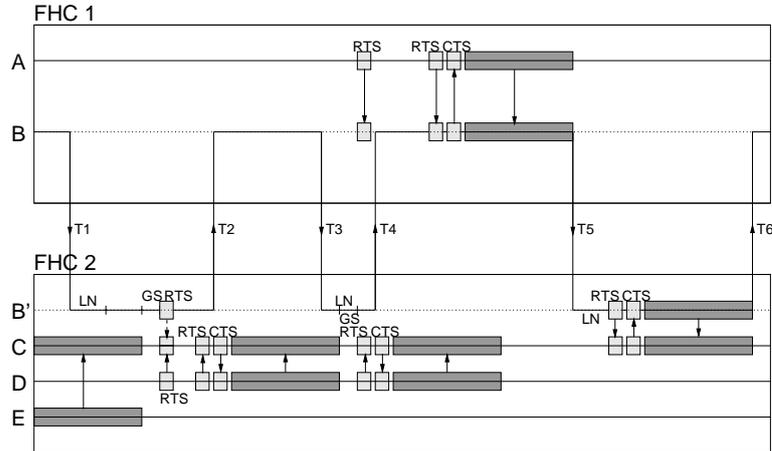


Figure 5: Example of Multiple Frequency Hopping Channel communication

When the destination node's FHC is different from the source node's home, then the source node has to switch between the source and destination FHCs during contention. This is illustrated in the figure, where node B wants to send a packet to node C in FHC 2. First,

it switches to FHC 2 (becomes B' after transition T1) and listens on the channel for at least a fixed amount of time (denoted by LN (listen) in the figure). This is needed to synchronize to the channel and determine if there is an ongoing data transmission in the FHC or not. If there is an ongoing data transmission, as in the example, then B must wait until this transmission is over (and observe the guard space, GS) before sending an RTS. In the figure, node D also wants to send to node C, and after colliding with B at the first RTS transmission, it wins the contention in the second stage. B notices this when it hears the RTS from node D and waits until this data transmission is over. For this period of time, it switches back to its home FHC (transition T2). To determine when it can try again with a new RTS, node B uses its estimate of the length of the data transmission given in the RTS packet (this information is also given in CTS packets). Node B switches back to FHC 2 (transition T3) such that it spends the period of LN before its backoff counter reaches zero. In the figure, node D wins the contention once again, and B switches back to FHC 1 (transition T4). In the meantime, node A initiates a data transmission to node B which is unsuccessful because node B is away at that time. The RTS is retransmitted later, and the subsequent data transmission is started to node B. This delays node B switching to FHC 2 once again. However, when the transmission in FHC 1 is over, node B can immediately switch to FHC 2 (transition T5). After a period of LN has passed and FHC 2 is sensed free, node B sends its RTS which is successfully received this time, allowing the consequent data packet transmission. Once this is over, node B switches back to its home FHC (transition T6).

The address and home FHC of a neighbouring node is known from a neighbour discovery mechanism. This is either based on a static configuration, or on beacon packets sent by the nodes [27]. Beacon packets can be sent at a dedicated frequency, or on a special frequency hopping sequence. In addition, beacon packets are sent on each FHC in order to synchronize the channel timing [24]. Note that while it is clear that we must ensure timing synchronization between two nodes that communicate with each other, MFHC does not require a network-wide synchronization mechanism.

I have validated the proposed MFHC communication scheme by implementing it in a packet-based simulator. The implementation proved the feasibility of the approach, and served as a tool for the performance analysis of the following subthesis.

### **Thesis 5.2 : Performance Analysis of MFHC Communication [J2], [D6.4-6.5]**

I have investigated the impact of multiple frequency hopping channel communication on the system's performance using analytical and simulation methods. In particular, I have compared the extreme case of MFHC, where each device has its own distinct FHC, to a reference case where the entire ad hoc network uses the same FHC. I have also analyzed a case where subsets of an ad hoc network form a partially closed communication group in the sense that members of one group communicate mostly with other members of the same group and rarely with other nodes of the ad hoc network. This scenario may be typical in some realistic ad hoc networks.

In the performance analysis, I have introduced a network and traffic model, and compared different FHC configurations. The primary performance metric is the total system throughput. To model a number of different application groups used over the same coverage area in an ad hoc networking scenario, I use a group-based traffic model: devices send most of their data to other members of the same group. In the numerical analysis, I consider the extreme case where nodes within a group send packets to the members of the same group only. The total of  $N$  nodes are divided into groups of size  $G$ . Sources are assumed to be greedy, which means that sources always have a packet to send. Before each packet transmission, the destination is chosen randomly and independently according to a uniform distribution from

the other nodes in the same group. Each of the  $N$  nodes are within transmission range of each other, so transmissions in different groups at the same time and same frequency collide.

In the analysis I distinguish the three different FHC configurations mentioned above based on the set of nodes that use a common home FHC for contention resolution and communication. In the *common* FHC case the same single channel is used by all of the  $N$  nodes. This serves as a reference case where devices do not need to switch to a different FHC. In the *device* FHC case there is a separate FHC for contention and data transfer for each device. In this case, for each destination a node has to switch to a new FHC. The third FHC configuration that I investigate represents a compromise between the two extremes. In the *group* FHC case, every group of  $G$  nodes has its own FHC for contention and data transmission. Since in our current traffic model packets are sent only within the group, therefore nodes do not have to switch to a different FHC in this case, either.

Figures 6 - 8 illustrate the three cases. The dark rectangles represent the data packets sent on a given hopping channel, while the lightly shaded rectangles represent contention on a given channel, with the arrows showing the direction of the data transmission and the contention.

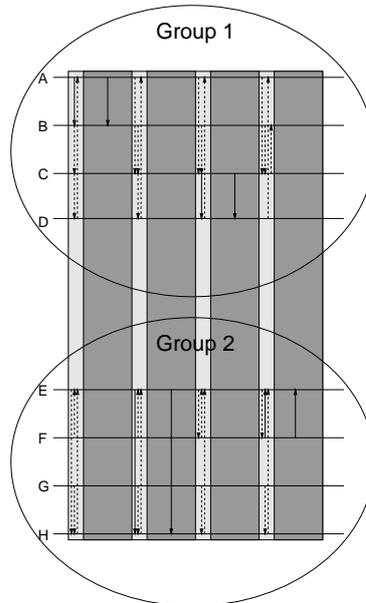


Figure 6: *Common* FHC: the same channel is used by all of the nodes.

- I have derived an analytical performance approximation of the three FHC configurations. The results are summarized in Table 2. Here,  $\lambda$  denotes the traffic offered by a node, that is, that fraction of time spent with transmission including retransmissions;  $\Lambda$  denotes load on an FHC;  $p$  denotes the packet loss probability;  $\Theta$  denotes total throughput;  $L_0$  denotes packet length;  $K$  denotes the number of hop frequencies.
- Based on the analytical results, I have made a performance comparison of the three considered FHC configurations with typical settings. I have made a number of observations. Below I also give some numerical figures using a given set of parameters [J2].

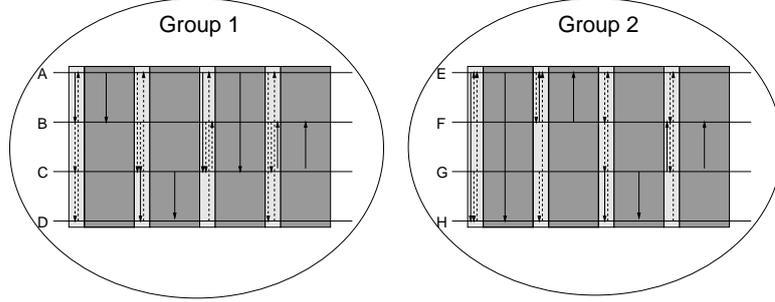


Figure 7: *Group* FHC: every group has its own channel.

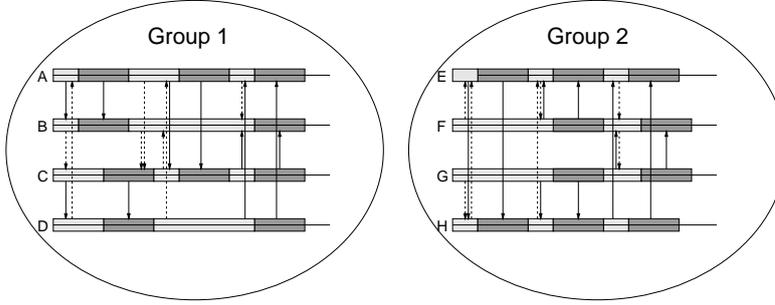


Figure 8: *Device* FHC: there is a separate channel for each of the nodes.

- In the *group* and *device* FHC configurations, per node offered traffic does not depend on the number of groups.
- The total throughput is constant for the *common* FHC case, and increases with the number of groups in the *device* and *group* cases. The *device* FHC configuration can give significantly higher total throughput than the *group* case (three times as much) due to its multiplexing gain.
- The *common* FHC can give the highest spectral efficiency if implemented in such a way that only a single channel is used in the network.
- For a group size of 2, the *group* FHC is more efficient than the corresponding *device* FHC configuration (e.g., by 15% depending on the parameters).

<i>Common</i>	<i>Group</i>	<i>Device</i>
$\lambda_c = \frac{1}{N} \left( \frac{L_0}{L_0+e} \right)$	$\lambda_g = \frac{1}{G} \left( \frac{L_0}{L_0+e} \right)$	$\lambda_d = \frac{L_0}{2L_0+\beta e}$
$\Lambda_c = \frac{L_0}{L_0+e}$	$\Lambda_g = \frac{L_0}{L_0+e}$	$\Lambda_d = \lambda_d$
$p_c = 0$	$p_g = 1 - \left( 1 - \Lambda_g \frac{2}{K} \right)^{N/G-1}$	$p_d = 1 - \left( 1 - \lambda_d \frac{2}{K} \right)^{N-1}$
$\Theta_c = \Lambda_c \left( 1 - \frac{1}{T_h} \right)$	$\Theta_g = \frac{N}{G} \Lambda_g (1 - p_g) \left( 1 - \frac{1}{T_h} \right)$	$\Theta_d = N \lambda_d (1 - p_d) \left( 1 - \frac{G}{T_h} \right)$

Table 2: Analytical performance results of three different FHC configurations

- For higher group sizes, the *device* FHC can provide higher throughput within a group than the *group* FHC (e.g., for a group size of 20, the *device* FHC can reach seven times as much throughput as the *group* FHC).
  - Shorter packets significantly decrease throughput efficiency due to the relatively higher overhead of contention (e.g., packets of 12 slots length can be twice as efficient as packets of 2 slots).
  - The *device* FHC configuration is the most sensitive to synchronization overhead, especially for higher group size. (E.g., for a group size of 10, the throughput degradation can reach or exceed 50% depending on the beacon period.)
- We have implemented the MFHC scheme in a packet-level simulator and carried out a simulation-based performance analysis. The simulator models interference in the physical layer taking into account the different frequency hopping patterns. Corrupted packets are retransmitted based on a one-bit ARQ feedback that is sent after each transmission. I have found that the simulation results reinforce the findings of the analytical performance model described above. Using simulations, I have been able to quantify the efficiency of contention in the *device* FHC case compared to the *group* FHC case and found that the decrease in efficiency of contention due to the fact that a node in the *device* FHC case has to switch between different channels can be as large as 8 to 16, depending on the parameters. Nevertheless, the *device* FHC configuration can still achieve significantly higher total throughput, as mentioned above, due to its additional multiplexing capability.
  - I have extended the traffic model to investigate the effect of traffic between the groups as well. In the extended model, with a probability  $p_{ng}$  a node chooses its destination from all the other nodes in the network, not just its own group. I have shown that the *device* FHC configuration is not sensitive to this change since it does not depend on the formation of groups. The *group* FHC shows a decrease in both small and large group sizes, but the decrease is much more significant when the group size is small (40%) than when it is large (25%). In both cases, the results show that the *device* FHC configuration gives higher performance in the case of heterogeneous traffic, that is, when there is significant traffic between the groups.
  - I have also investigated the effect of changing the traffic pattern within a group to model a client-server application (with no traffic between the groups). In this case I designate one node in all groups to be a server and the other nodes in the group to be clients. All nodes remain greedy as before in that they always have a data to send, but with a constant probability  $p_s$ , the clients choose the server as their destination. In this experiment the total throughput of the *group* FHC configuration remains constant since this is determined by the capacity of the group channel. On the other hand the performance of the *device* FHC configuration decreases to that below the *group* case. When there is only server-client traffic, the *device* FHC case can not achieve multiplexing gain, and it is uses a less efficient contention scheme than the *group* FHC which explains its lower performance.

## 5 Application of New Results

Thesis 1 can be applied in the characterization of traffic on data networks. Currently we have little knowledge about the nature of traffic in the internet and how it changes with changing

user demand and new applications. The results of this thesis can be applied to measurements taken in commercial networks. Such measurements and analysis could lead to more efficient network design.

The proposal in Thesis 2 has become part of the HIPERLAN/2 Wireless LAN standard specified by the BRAN (Broadband Radio Access Networks) project of ETSI (European Telecommunications Standards Institute) [28]. The standard was approved in 1999 [18]. The system works in the 5GHz and provides local high-speed connectivity with data rates up to 54 Mbps using OFDM at the physical layer [19, 20, 21]. The ARQ proposal was discussed in a number of standardization contributions [S1]-[S6]. Since the SRPB was accepted in the final standard, it will be applied in all products that conform to the requirements of the HIPERLAN/2 specification. Among other potential vendors, a prototype has been implemented through a combination of software and hardware features by Ericsson.

Various aspects of the protocol have been patented [P1, P2, P3] by Ericsson. In addition, the work in Thesis 3 and Thesis 4 have been used as inputs in the implementation of wireless LAN and other systems.

The proposed architecture of Thesis 5 has also been patented [P6, P7] by Ericsson. It is readily applicable for implementation using low-cost frequency hopping radios such as Bluetooth. Other solutions related to the research described in Thesis 5 have also been patented [P4, P5, P8, P9].

## References

- [1] Leonard Kleinrock. *Queueing Systems. Vol I: Theory*. John Wiley & Sons, 1975.
- [2] Vern Paxson and Sally Floyd. Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.
- [3] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1), February 1994.
- [4] Sándor Molnár and Attila Vidács. On modeling and shaping self-similar ATM traffic. In *Proceedings of ITC-15, Washington D.C., USA*, June 1997.
- [5] Raif O. Onvural. *Asynchronous Transfer Mode Networks, Performance Issues*. Artech House, Boston, London, 1994.
- [6] Kotikalapudi Sriram and Ward Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, 4(6), September 1986.
- [7] R. Gusella. Characterizing the variability of arrival processes with indexes of dispersion. *IEEE Journal on Selected Areas in Communications*, 9(2), February 1991.
- [8] A. E. Eckberg, Jr. Generalized peakedness of teletraffic processes. In *ITC-10*, Montreal, 1983.
- [9] A. E. Eckberg, Jr. Approximations for bursty (and smoothed) arrival queueing delays based on generalized peakedness. In *ITC-11*, Kyoto, Japan, 1985.
- [10] G. D. Stamoulis, M. E. Anagnostou, and A. D. Georgantas. Traffic source models for ATM networks: a survey. *Computer Communications*, 17(6), 1994.
- [11] Sándor Molnár, István Cselényi, and Nils Björkman. ATM traffic characterization and modeling based on the leaky bucket algorithm. In *IEEE Singapore International Conference on Communication Systems*, Singapore, November 1996.
- [12] H. Heffes and J. M. Holtzman. Peakedness of traffic carried by a finite trunk group with renewal input. *The Bell System Technical Journal*, 52(9):1617–1642, November 1973.
- [13] Brian L. Mark, David L. Jagerman, and G. Ramamurthy. Peakedness measures for traffic characterization in high-speed networks. In *Proceedings of IEEE INFOCOM'97*, 1997.
- [14] Demetres Kouvatsos and Rod Fretwell. Batch renewal process: Exact model of traffic correlation. In *High Speed Networking for Multimedia Application*, pages 285–304. Kluwer Academic Press, 1996.
- [15] Broadband radio access networks (bran) HIPERLAN type 2 data link control (dlc) layer part 1: Basic data transport functions, April 2000.
- [16] Hui Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10), October 1995.
- [17] Jon C. R. Bennett and Hui Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689.
- [18] HiperLAN/2 Global Forum. <http://www.hiperlan2.com/>.
- [19] Martin Johnsson. HiperLAN/2 — the broadband radio transmission technology operating in the 5ghz frequency band, 1999. White Paper of the HIPERLAN/2 Global Forum.

- [20] Jamshid Khun-Jush, Gran Malmgren, Peter Schramm, and Johan Torsner. HIPERLAN type 2 for broadband wireless communication. *Ericsson Review*, 2000.
- [21] Jamshid Khun-Jush, Gran Malmgren, Peter Schramm, and Johan Torsner. Overview and performance of HIPERLAN type 2 - a standard for broadband wireless communications. In *Proceedings of VTC2000 Spring*, Tokyo, Japan.
- [22] S. Corson and J. Macker. Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations, 1999. RFC2501.
- [23] Bluetooth specification 1.1.
- [24] IEEE. IEEE 802.11 family of standards.
- [25] Zhenyu Tang and J. J. Garcia-Luna-Aceves. Hop-reservation multiple access (HRMA) for ad-hoc networks. In *IEEE infocom*, Mar 1999.
- [26] A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 1987.
- [27] Miklós Aurél Rónai and Eszter Kail. A simple neighbour discovery procedure for Bluetooth ad hoc networks. In *Proceedings of Globecom*, December 2003.
- [28] ETSI Broadband Radio Access Networks project. <http://www.etsi.fr/bran/>.

## Publications

### [J] Journal Papers

- [J1] Gy. Miklós, “Bluetooth: Olcsó, drótnélküli helyi összeköttetés (Bluetooth: cheap wireless local connectivity),” *Magyar Távközlés*, vol. 11, No. 9, pp. 11-14, September 2000 (in Hungarian).
- [J2] Gy. Miklós, F. Kubinszky, A. Rácz, Z. R. Turányi, A. G. Valkó, M. A. Rónai, S. Molnár, “A Novel Scheme to Interconnect Multiple Frequency Hopping Channels into an Ad Hoc Network,” *ACM Mobile Computing and Communications Review (MC2R)*, Special Issue on Wireless PAN & Sensor Networks, vol. 8, No. 1, pp. 109-124, January 2004.

### [C] Conference and Workshop Papers

- [C1] S. Molnár, Gy. Miklós, “On Burst and Correlation Structure of Teletraffic Models,” 5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, 21-23 July 1997, Ilkley, U.K.
- [C2] S. Molnár, Gy. Miklós, “Peakedness Characterization in Teletraffic,” IFIP TC6, WG6.3 conference, Performance of Information and Communications Systems, PICS’98 Lund, Sweden, May 25-28, 1998.
- [C3] Gy. Miklós, S. Molnár, “Fair Bandwidth Allocation of a Wireless Base Station,” First Workshop on IP Quality of Service for Wireless and Mobile Networks (IQWiM99), April 1999, Aachen, Germany.
- [C4] Gy. Miklós, S. Molnár “Fair Allocation of Elastic Traffic for a Wireless Base Station,” IEEE GLOBECOM’99, December, 1999 Rio de Janeiro, Brazil.
- [C5] H. Li, J. Lindskog, G. Malmgren, Gy. Miklós, F. Nilsson, G. Rydneil, “Automatic Repeat Request (ARQ) Mechanism in HIPERLAN/2,” VTC2000 Spring, May 2000, Tokyo, Japan.
- [C6] Gy. Miklós, S. Molnár “Distributed Fair Bandwidth Allocation of a Wireless Base Station,” Networking 2000, May 14-19, 2000, Paris, France. In: G. Pujolle et al. (Eds.), “Lecture Notes in Computer Science 1815,” Springer-Verlag Berlin Heidelberg 2000, pp 689-701.
- [C7] Gy. Miklós, S. Molnár, “Analysis of a Distributed Wireless Fair Scheduling Scheme,” ITC Specialist Seminar on Mobile Systems and Mobility, pp 301-312, March, 2000, Lillehammer, Norway.
- [C8] Gy. Miklós, A. Rácz, Z. Turányi, A. Valkó, P. Johansson “Performance Aspects of Bluetooth Scatternet Formation,” Proceedings of MobiHoc 2000, pp. 147-148, Boston, USA.
- [C9] A. Rácz, Gy. Miklós, F. Kubinszky, A. Valkó, “A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets,” Proceedings of MobiHoc 2001, pp. 193-203, Long Beach, USA.

### [T] Technical Reports

- [T1] Gy. Miklós, “Peakedness Measures,” technical report, High Speed Networks Laboratory, Technical University of Budapest, 1997.
- [T2] S. Molnár, Gy. Miklós, “Generalized Peakedness in Discrete Time,” COST 257 TD(98)052, Granada, Spain, September 24-25, 1998.
- [T3] S. Molnár, Gy. Miklós, “A Simple Compensation Mechanism for Wireless Resource Allocation,” COST 257 TD(99)037, September 30 - October 1, 1999, Larnaca, Cyprus.

### [S] Standardisation contributions

- [S1] Ericsson, “Selective repeat ARQ mechanism in HiperLAN2,” HL12.5ERI7a, March 1999, Stockholm, Sweden, contribution to ETSI BRAN standardisation.
- [S2] Ericsson, “ARQ message formats,” HL13ERI8a, April 1999, Stockholm, Sweden, contribution to ETSI BRAN standardisation.
- [S3] Ericsson, “Flow control in HiperLAN 2 ARQ protocol,” HL14ERI11a, June 1999, Sophia Antipolis, France, contribution to ETSI BRAN standardisation.
- [S4] Ericsson, “Polling bit in FCCH RG for RR,” HL14ERI20a, June 1999, Sophia Antipolis, France, contribution to ETSI BRAN standardisation.
- [S5] Ericsson, “Discard mechanism in the HiperLAN2 ARQ protocol,” HL14ERI22b, June 1999, Sophia Antipolis, France, contribution to ETSI BRAN standardisation.
- [S6] Ericsson, “Error handling in the HiperLAN2 ARQ protocol,” HL14ERI24a, June 1999, Sophia Antipolis, France, contribution to ETSI BRAN standardisation.
- [P] **Patents**
- [P1] Gy. Miklós, “Discard Mechanism for Selective Repeat Automatic Repeat Request,” US patent application 09/273259, March 1999.
- [P2] J. Lindskog, J. Ollson, G. Malmgren, Gy. Miklós, “Ordering of LCH&SCH,” US patent application 09/544219, April 2000.
- [P3] G. Rydneil, J. Lindskog, G. Malmgren, Gy. Miklós, H. Li, F. Nilsson, “ARQ C-PDU Format,” US patent application 60/128041, April 2000.
- [P4] Gy. Miklós, M. Félegyházi, “Traffic Dependent Bluetooth Scatternet Optimization Procedure,” US patent application No. 09/666529, September 2000.
- [P5] Gy. Miklós, Z. Turányi, A. G. Valkó, “Flexible Neighbour Discovery in Bluetooth,” UK patent application No. 0110397.7, April 2001.
- [P6] Gy. Miklós, Z. Turányi, A. G. Valkó, “Lightweight Piconet Concept in Bluetooth,” UK patent application No. 0110396.9, April 2001.
- [P7] Gy. Miklós, Z. Turányi, A. G. Valkó, F. Kubinszky, A. Rác “Random Access Communication in Bluetooth,” UK patent application No. 0110399.3, April 2001.
- [P8] A. Rác, Gy. Miklós, A. G. Valkó, F. Kubinszky, “Signaling Free, Self Learning Scatternet Scheduling Using Checkpoints,” US patent application No. 09/840241, April 2001.
- [P9] Gy. Miklós, A. Rác, “Predictable Communication Establishment in Ad-hoc Wireless Network” US patent application No. 10/039606, October 2001.
- [P10] Gy. Miklós, A. Rác, “Fast Hard Handover” WIPO patent application No. PCT/SE2004/000192, February 2004.