We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Fuzzy Based Flow Management of Real-Time Traffic for Quality of Service in WLANs

Tapio Frantti and Mikko Majanen
VTT Technical Research Centre of Finland
Finland

1. Introduction

Designing heterogeneous bandwidth limited communication systems that support a wide variety of applications, including file transfer, web browsing, interactive games, audio and video calls, and emerging real-time virtual world and social media applications is a challenging task because there is a shortage of resources to satisfy all traffic demands and diverse quality of service (QoS) requirements. For example, the current Internet architecture supports only best-effort service class which is not enough especially for delay sensitive real-time multimedia applications. Therefore, to improve QoS for specified traffic in the Internet, the end nodes (hosts) should make a bandwidth reservation through all the intermediate nodes, like access points and routers, by using some sort of resource reservation. For the QoS guarantee, the IETF has worked on the resource reservation protocol (RSVP) that can be used to hard resource reservation: an endpoint uses RSVP to request a simplex flow through the network with specified QoS bounds and the intermediate nodes, like routers, along the path either agree to honor the request or deny it. It is a transport layer protocol designed to reserve resources across a network. RSVP operates over an internet protocol versions 4 or 6 (IPv4 or IPv6) and provides receiver-initiated setup of resource reservations for multicast or unicast data flows. The drawback of the RSVP is that all the routers along the path must agree the resource reservation for QoS guarantee. However, no any QoS system can satisfy all users' demands if the network traffic exceeds network capacity. Another disadvantage is that the reserved virtual links do not necessarily use the network capacity optimally. Therefore, we focus here to the cognitive flow management of delay sensitive constant bit rate real-time traffics, such as voice over internet protocols (VoIP), video calls, and interactive games, to improve QoS in Wireless Local Area Networks (WLANs).

The Internet has two independent flow problems. Internet protocols need end-to-end *flow control* and a mechanism for intermediate nodes, like routers and access points, to control the amount of traffic known as the *congestion prevention and control* mechanism. *Flow control* is closely related to the point-to-point traffic between a sender and a receiver. It guarantees that a fast sender cannot continually send datagrams faster than a receiver can absorb them. *Congestion* is a condition of severe delay caused by an overload of datagrams at intermediate nodes. Usually congestion arises for two different reasons: a high-speed computer may be able to generate traffic faster than a network can transfer it or many computers send datagrams simultaneously through a single router, even though no single computer causes the problem. Hence, the congestion control can be considered more as a global issue whereas

flow control is more a local, point to point, issue with some direct feedback from the receiver to the sender.

The term *cognition* refers to the processing of information, applying knowledge, and changing preferences. In the communication networks, cognition can be used to improve the performance of resource management, quality of service, security, control algorithms, or many other network goals. Here we define *cognitive flow management* as a cognitive process that can perceive current network conditions, and then plan, decide, and act on those conditions for improved quality of service.

In our earlier publications (Frantti & Majanen, 2010; Frantti et al., 2010) we presented and compared PID (Proportional, Integral, Derivative) and fuzzy control systems, which adjust packet size of UDP (User Datagram Protocol) based uni- or bidirectional traffic flow on WLANs according to prevailing channel conditions. They aimed to optimize packet sizes of real-time traffic flows for the prevailing connection for higher end-to-end throughput by fulfilling the overall application dependent delay requirement. In this chapter, the *aim of the flow management system is to adjust appropriate packet size and transmission interval of the source node's constant bit rate traffic flows for prevailing network conditions to achieve application dependent quality of service requirements*. Hence, the research question can be stated here as follows: *"How to manage constant bit rate real-time traffic flows so that application dependent quality of service requirements are achieved with the optimal network capacity?"*. Although the main goal of this work is related to the quality of service of WLAN systems and the simulations and results were performed for the IEEE 802.11*b* system, the approach and the techniques are not limited to these systems, but are easily applicable to other packet switched networks as well.

The organization of the rest of the chapter is as follows. Section 2 presents a literature review of the weak resource reservation and quality of service in communication networks. It also presents a review of the packet size optimization in wireless networks. Section 3 briefly summarizes the structure and channel access of the WLANs. Section 4 introduces the principles of service classification whereas Section 5 gives an introduction to weak resource reservation, like congestion prevention and control, flow control and denying and/or degrading services and reduction of channel access competition by admission control. In Sections 7 and 8 are briefly summarized the basic principles of the developed PID and fuzzy system based controllers. Section 9 depicts the developed simulation model and simulation scenarios. Section 10 comprises achieved results with the controllers. Finally, conclusions are presented in Section 11 .

2. Literature review

2.1 Hard resource reservation

For the QoS guarantee, the IETF has worked on the transport layer protocol called resource reservation protocol (RSVP) that can be used to hard resource reservation across a network. Integrated Services is often associated with RSVP. The Integrated Services architecture divides the flows to different service classes (e.g. guaranteed service class for intolerant applications that require that a packet never arrives late), and then RSVP is used for reserving the needed resources for each service class.

2.2 Weak resource reservation: packet scheduling and queueing methods

Weak resource allocation schemes without actual reserved virtual links closely includes packet scheduling schemes and queueing methods (Kleinrock, 1975). The queueing algorithm can be thought of as allocating bandwidth to packets on the intermediate nodes. The most popular queueing algorithm is First-In-First-Out (FIFO), which determines the service order of packets

based on their arrival order. In Priority Queueing (PQ), traffic classes with the highest priority are forwarded with the least delay (Huitema, 2000; Nagle, 1987; Sanjay & Hassan, 2002). In Class Based Queueing (CBQ) traffic classes are forwarded with equal share (Floyd & Jacobson, 1995), *e.g.*, Round Robin (RR) algorithms process packets in turn with equal share and achieve very high accuracy and fairness in the output bandwidth sharing but cannot provide tight delay guarantees (Nagle, 1985). In Fair Queueing (FQ) techniques, like the Weighted Fair Queueing (WFQ), are assigned a weight to each output queue (Demers et al., 1989). However, scheduling and queueing methods provide a rather weak form of resource reservation and cannot guarantee QoS, because weights are only indirectly related to the bandwidth the flow receives. The another problem of these methods and their modifications is that they are quite static in their operations. The latest development of scheduling methods is directing to the dynamic adaptation of scheduling parameters which gives better overall performance. There exists some related articles such as (Crawford & Marshall, 2001; Horng et al., 2001; Sayenko et al., 2006; 2003) devoted to the adaptive WFQ. In Horng et al. (2001) the developed adaptive approach to WFQ is a variation of fair queue algorithm with dynamic priority scheduling. An adaptive approach to WFQ that uses a concept of revenue to adapt weights is presented in Sayenko et al. (2003). This adaptive WFQ algorithm is later extended in (Sayenko et al., 2006) to an comparison and analysis of several adaptive scheduling algorithms: Revenue-based adaptive WFQ (RA-WFQ), revenue-based adaptive Weighted Round Robin (RA-WRR) and revenue-based adaptive Deficit Round Robin (RA-DRR). In Crawford & Marshall (2001) a new fast packet scheduling algorithm called Dynamic Weighted Fair Queuing (DWFQ) is created. We have considered in our previous publication fuzzy expert systems for adaptive weighted fair queueing and service classification (Frantti & Jutila, 2009).

2.3 QoS in wireless networks

Wireless network protocols are designed based on a layered approach, where each layer in the protocol stack is designed and operated independently. The interfaces between layers are rather static. There are many studies that examine QoS provisioning in wireless networks with a layered perspective, concentrating only on one layer at the time, *e.g.* on power control or modulation/rate adaptation on the physical layer, scheduling or channel access on the MAC layer, admission control or routing on the network layer, rate or congestion control on the transport layer, or video and image coding schemes on the application layer. Perkins & Hughes (2002) includes a survey of QoS support for wireless mobile ad hoc networks including QoS routing protocols, resource reservation schemes, and QoS aware MAC layers. QoS aware MAC layers for wireless ad hoc networks are also reviewed in Kumar et al. (2006). However, strict layered design is not optimal for wireless multihop networks because of their dynamic nature. In wireless networks the layers should cooperate more closely to jointly optimize the overall performance, especially in case of real-time applications with high bandwidth and/or stringent delay requirements. Many studies, *e.g.* (Goldsmith & Wicker, 2002; Huusko et al., 2007; Lamy-Bergot et al., 2010; Qu et al., 2005; Setton et al., 2005), on wireless networks show that a cross-layer design can significantly improve the system performance. A cross-layer approach seeks to enhance the performance of a system by breaking the independence of the layers by jointly designing multiple protocol layers. Zhang & Zhang (2008) surveys multiple possibilities for cross-layer interactions in wireless multihop networks.

Fuzzy set theory has also been used for enhancing the QoS in wireless networks. For example, authors in (Khoukhi & Cherkaoui, 2008) present a fuzzy decision support system for wireless ad hoc network. They use fuzzy set theory for best-effort traffic regulation, and propose

schemes for real-time traffic regulation, and admission control. Chan et al. (2001) apply fuzzy set theory to employ decision criteria such as user preferences, link quality, cost, or quality of service (QoS) for handover decision scheme.

2.4 Packet size optimization for connection quality

Korhonen & Wang (2005) have studied the effect of packet size on loss rate and delay in IEEE 802.11 based WLAN. The analysis shows that there is a straightforward connection between bit error characteristics and observed delay characteristics. This information can be useful in adjusting application level framing under different network conditions. For example, an intelligent streaming application could optimize end-to-end delay and wireless resource utilization by analyzing the delay pattern for packets with different lengths. In general, it is shown throughout the literature that the performance of wireless networking is sensitive to the packet size, and that significant performance improvements are obtained if a "good" packet size is used. For example, authors in (Bakshi et al., 1997) show this for TCP traffic over wireless network. Chee & David (1989), Lettieri & Srivastava (1998), and Chien et al. (1999) do study of the relationship between frame length and throughput, but they do not propose any exact method to dynamically control the frame length. Packet size optimization has been studied also in several other perspectives, like energy efficiency in (Sankarasubramaniam et al., 2003) and security in (Younis et al., 2009), but these solutions are statistical in nature, meaning that the packet size is optimized beforehand. Work done in (Smadi & Szabados, 2006) is somehow related to our work, but even in this article the focus is different, error recovery in communication rather than optimal packet size in the first place. PLFC (Sheu et al., 2000) is the most similar to our approach presented in this chapter. PLFC is a fuzzy packet length controller for improving the performance of WLAN under the interference of microwave oven. The input parameters for the fuzzy controller are the packet length and the packet error rate. It is shown that PLFC improves the throughput of UDP traffic compared to using fixed length packets.

In the most recent of our publications (Frantti & Majanen, 2010; Frantti et al., 2010) we presented and compared PID and fuzzy control systems, which adjust packet size of UDP based uni- or bidirectional traffic on WLANs according to prevailing channel conditions. In other words, (Frantti & Majanen, 2010; Frantti et al., 2010) considered flow control for a fixed delay requirements. The delay can be defined as the time taken by a packet to traverse the network. Here the aim of the flow management system is to achieve quality of service requirements of the real-time applications with the optimal network capacity. Hence, the control system adjusts appropriate packet size and transmission interval of the source node's real-time traffic flows for the maximum number of such real-time connections as VoIP calls, video calls, and interactive games.

3. Wireless local area network

The market for wireless communications has grown rapidly since the introduction of the 802.11*b*, *g*, and *a* WLAN standards offering performance almost comparable to the Ethernet. The 802.11*b*, *g*, and *a* standards specify the lowest (physical) layer of the OSI reference model and a lower part (MAC) of the next higher layer (data link layer). The standards specify also the use of the 802.2 link layer control protocol, which is the upper portion of the data link layer.

The IEEE 802.11*b* wireless local area networks use the 2.4 GHz ISM (Industrial, Science and Medical) license-free frequency band, which is divided into 11 usable channels. Any particular network can use only less than half of these in operation, but all network hardware is built to

be able to listen to and transmit on any of the channels. The sender and receiver must be on the same channel to communicate with each other.

The IEEE 802.11*b* network can be set to work in an Independent Basic Service Set (IBSS), in a Basic Service Set (BSS) or in an extended service set (ESS) mode. The IBSS is an ad hoc group of independent wireless nodes which communicate on a peer-to-peer basis. A standard refers to a topology with a single access point as a BSS. The arrangement with multiple access points is called an ESS (B. Bing, 2002). In ESS nodes transmit data to the nearest access point, which delivers it either to another node in the coverage area or to some other node(s) on the Internet. In WLANs nodes can transmit only when a communication channel is unoccupied. The channel access is regulated by media access control (MAC) protocols, which are typically contention-based protocols. The IEEE 802.11*b* MAC supports two modes of operation: the Point Coordination Function (PCF) and the Distributed Coordination Function (DCF). The PCF provides contention free access, while the DCF uses the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism for contention based access. Here we consider only DCF mode, because PCF mode is not commonly used and it is not a part of, *e.g.,* the Wi-Fi Alliance's interoperability standard (Leung et al., 2002; Li & Ni, 2005).

In contention-based MACs, the transmission bursts intervals for nodes are irregular (transmission jitter) and vary according to the type of transmitted traffic and the number of nodes competing or reserving the channel. The packet interval is also dependent on the packet length. Therefore, the packet transmission interval and the channel access time are decreased, when the packet size is reduced. This increases channel reservation competition and may lead to the network congestion and decreased throughput of the network. On the other hand, when the packet payload is increased, the number of packets sent from the source node is reduced and the packet interval becomes longer. Then the channel is free for a longer period of time between packets, which reduces the channel reservation competition and increases the probability of getting a free channel. However, when the packet size increases the bit errors caused by the channel increase the probability of a packet error, which increases packet loss and decreases throughput. The channel access time depends on also the type of traffic exchange. For example, in connection-oriented communication also acknowledgement (ACK) frames have to compete the channel access time in reverse direction, which decreases network node's channel access time in forward direction, too.

The IEEE 802.11*e* defines a set of QoS enhancements for WLAN applications. It was included in the 802.11-2007 standard together with amendments *a*, *b*, *d*, *g*, *h*, *i*, and *j* in July 2007. Instead of PCF and HCF, 802.11*e* defines HCF Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA). Both HCCA and EDCA defines Traffic Categories (TC), which can be used for separating voice, video, best effort, and background traffic from each other.

In EDCA, shorter contention window (CW) and arbitration inter-frame spacing (AIFS) are used for higher priority traffic packets. As a result, higher priority packets are sent a little bit earlier on average than lower priority packets during contention periods. EDCA has also contention-free periods called Transmit Opportunity (TXOP). A TXOP is a bounded time interval during which a station can send as many frames as possible as long as the duration of the transmissions does not extend beyond the maximum duration of the TXOP. For voice and video traffic, the maximum duration of the TXOP is greater than for other type of traffic. Wi-Fi Multimedia (WMM) certified APs must be enabled for EDCA and TXOP.

HCCA works pretty similar to PCF. However, in contrast to PCF, in which the interval between two beacon frames is divided into two periods of CFP and CP, the HCCA allows AP to initiate CFP almost anytime to send or receive a frame to or from a station in contention-free

manner. During a contention-free periods the AP controls the access to the medium. During the contention periods, all stations function in EDCA. In addition to Traffic Classes (TC), HCCA defines also Traffic Streams (TS), which allows a sort of per-session service instead of per-station queuing. AP can coordinate these streams in any fashion it chooses. This makes HCCA perhaps the most complex coordination function, but on the other hand, HCCA allows the QoS to be configured with great precision. For example, QoS-enabled stations may request some specific QoS parameters (data rate, jitter, etc.), which should allow advanced applications like VoIP and video streaming to work more effectively. HCCA support is not mandatory in WMM certified APs.

4. Service classification

4.1 QoS parameters

The term QoS itself refers to statistical performance guarantees that a network can make. Typical QoS parameters can be categorized to cost, format, performance, synchronization and user classes. Cost parameters include costs of connection and data transfer. Compression, frame rate, and resolution are format parameters. Bit rate and delays are typical performance parameter whereas skews in multimedia transmission is an example of synchronization parameters. User parameters are, for example, subjective voice and quality of image. It is up to transport layer to examine the parameters, and determine whether it can provide the required service. The typical transport layer QoS parameters are: connection establishment delay and failure probability, throughput, transit delay, residual error ratio, protection, priority and resilience (Tanebaum, 1996).

4.2 Service categories

Due to rich space of application requirements, a richer service model than best-effort service is needed to meet the need of applications. This leads to to a service model with more than just the best-effort class, each class available to meet the needs of some set of applications. There are two broad categories developed to provide a range of qualities of service: *fine-grained* and *coarse-grained* approaches. Fine-grained approaches provide QoS to individual applications or flows whereas coarse-grained approaches provide QoS to large classes of data or aggregared traffic.

Integrated Services, which is a QoS arhitecture developed in the IETF (Internet Engineering Task Force) and often associated with RSVP (Resource Reservation Protocol) is an example of the fine-grained approches. The Integrated Services architecture allocates resources to individual flows. The IETF IntServ working group developed specifications of a number of service classes, such as guaranteed service and controlled load, designed to meet the needs of some of the application types. It also defined how to use RSVP to make reservations using these service classes. Guaranteed service class is designed for intolerant applications, which require that a packet never arrive late. The network should guarantee that the maximum packet delay has some specified value. Controlled load service class is aimed to meet the needs of tolerant, adaptive applications. Tolerant applications run quite well on networks that are not heavily loaded. The aim of the controlled load service is to emulate a lightly loaded network for those applications that request the service, even though the network as a whole may in fact be heavily loaded. The trick to this is to use a queuing mechanism, such as weighted fair queuing to isolate the controlled load traffic from the other traffic (Peterson & Davie, 2007).

In the coarse-grained category lies, for example, perhaps the most widely used QoS mechanism *Differentiated Services*. The Differentiated Services allocates resources to a small

number of traffic classes. Many proposed Differentiated Services approaches simply divide traffic into two classes. The purpose is to add the service model in small increments in order to avoid difficulties that network operators already experience just trying to keep a best-effor internet running smoothly (Peterson & Davie, 2007).

In this work the aim of the flow management system is to achieve quality of service requirements of the real-time applications for the maximum number of such real-time connections as VoIP calls, video calls, and interactive games.

5. Weak resource reservation

In this chapter the resource allocation schemes without actual reserved virtual links is referred as a weak resource allocation. It closely includes packet scheduling schemes and queueing methods, congestion control and prevention, admission control and flow control.

5.1 Scheduling and queueing

One main tool for implementing network QoS are the intelligent scheduling and queueing algorithms. Queueing algorithms participate in congestion control and prevention and for allocating resources. In congestion prevention, routers monitor the output lines and allocate resources for different applications efficiently. Powerful resource allocation to individual traffic flows is closely in conjunction with choosing the right kind of packet scheduler. If there is a situation that network resources cannot serve all flows, queues will start to build up in routers. A packet scheduler is in important role in dequeueing the packets and keeping track of the network resources. In datagram-based Internet all the resources are shared on a per-packet basis compared to the traditional circuit-switched telephone system where all flows are completely isolated from each other. If there is a shortage of resources to satisfy all traffic demands, bandwidth must be shared fairly to all competing flows.

Queueing disciplines can be classified into work-conserving and non-work-conserving (Wang, 2001). Work-conserving discipline always schedules packets when there are packets waiting for service in the queue. Most of the well-known schedulers are work-conserving. However, non-work-conserving algorithms are also competent because they are proposed to reduce jitter and buffer size in the network while they only schedule packets that are considered to be eligible.

The most popular queueing algorithm is the First-In-First-Out (FIFO) which determines the service order of packets strictly based on their arrival order. In Priority Queueing (PQ) (Nagle, 1987), traffic classes with the highest priority are forwarded with the least delay. The drawback of PQ algorithms is that packets with lower priority can suffer from unfair service treatment. Round Robin (RR) algorithms (Nagle, 1985) and its extensively used versions Weighted Round Robin (WRR) (Hahne, 1986) and Deficit Round Robin (DRR) (Shreedhar & Varghese, 1995) process packets in turn with equal share. RR scheduling techniques cannot achieve very good accuracy and fairness when sharing the output bandwidth. Another drawback is that RR algorithms are not able to provide tight delay guarantees. These problems were defeated with Fair Queueing (FQ) techniques (Demers et al., 1989) of which the Weighted Fair Queueing (WFQ) is no doubt the most popular and studied one. Several commercial router and switch vendors are implementing WFQ in their products.

5.2 Congestion prevention and control

For the Internet congestion and resource control has been a research challenge for a long time. Congestion occurs when the aggregate demand for a resource exceeds the available capacity of the resource, *i.e.*, congestion conditions occur when a network cannot handle all the traffic that

is offered. An increase of the offered load does not necessarily imply an increase of throughput but it may even happen in congestion condition that the throughput is reduced as the offered load increases which may due to, *e.g.*, the aggressive retransmission techniques used by some network protocols to compensate packet loss. Resulting effects include long delays, wasted resources due to lost or dropped packets, or even possible congestion collapse, in which all communications in the entire network ceases. Therefore, it is evident that certain mechanisms is required to maintain good network performance and to prevent the network from being congested.

For the congestion handling there are two main approaches, namely *congestion control* and *congestion prevention*. Congestion control is a reactive method and comes into play after the network is overloaded. Congestion control involves the design mechanisms to limit the demand-capacity mismatch and dynamically control traffic sources when such a mismatch occurs. Especially for real-time traffic, it is important to understand how congestion arises and find efficient ways to keep the network operating within its capacity. The basic design issues of the congestion control are what to feedback to sources and how to react to the feedback. However, endpoints, *i.e.*, the source and destination do not usually have the details of congestion point(s) and reason(s). Intermediate nodes, on the other hand, can use network layer techniques like ICMP (Internet Control Message Protocol, one part of the Internet protocol family) to inform hosts that congestion has occured.

The most widely used congestion control mechanisms are *drop-tail*, *active queue management*, *DECbit mechanism*, *random early detection* and it's numerous variants, *explicit congestion notification*, and *partial buffer sharing*. *Drop-tail* works on first-in-first-out queue, which drops incoming packets when the queue becomes full. *Active queue management* detects congestion and acknowledges the sources about it before queue gets overflow. *DECbit mechanism* is based on the congestion notification bit in the packet header. It provides feedback to the sources for flow control. In *random early detection* incoming packets are dropped probabilistically before the queue becomes full. *Explicit congestion notification* extends random early detection in a way that instead of dropping a packet it marks it when the average queue size lies between specific threshold values. *Partial buffer sharing* scheme controls the allocation of buffer to various traffic classes with the delay constraints to meet diverse QoS demands. Interested reader finds more information about the congestion control mechanisms, for example, from (Ahmad et al., 2009).

Congestion prevention is a proactive approach and it acts before the network is overloaded, *i.e.*, it plays a major role before the network faces congestion. Congestion prevention aims to reduce congestion by designing good protocols and it takes proactive actions without relying on the network status. Congestion prevention covers different policies at the transport, network, and data link layer such as retransmission, acknowledgement, flow control, admission control, and routing algorithm. The end systems typically negotiate with the network and after that systems act independently. The end-systems get no information from the network about the current traffic and network status. However, in wireline networks intermediate nodes, such as routers, can monitor their output lines' load. Hence, whenever the utilisation of a line approaches a specified threshold level, the router transmits *choke* datagrams to the sources in order to give warning signals to them. The source nodes or hosts are required to reduce transmission rate to the specified destination by n percentage. Another paradigm that has been suggested for use in congestion prevention is *weighted fair queuing*, where a router selects datagrams from multiple queues in a round robin way to the idle output line. The router weights more bandwidth to some services than others. In packet switched networks it is also possible to allow new virtual circuits by routing traffic via a different, uncongested, route. Another alternative solution is to negotiate an agreement

between the hosts and network during the connection set up by specifying the volume and the shape of the traffic as well as quality of service requirements.

If congestion does not disappear with the preventive actions, routers can throw away datagrams they cannot handle (*load shedding*). They can do it either randomly or in a rational way, for example, when dropping a file transfer, a newer one is more rational than an older one due to acknowledgement and retransmission procedures. On the contrary, in real-time data transfer newer ones are more valuable than older ones. In congestion prevention it is also suggested to use media access layer solutions, like decreasing excessive overhead, retransmissions and auto-rate fallback.

5.3 Admission control

In wireless networks, admission control and resource reservation mechanisms are commonly proposed for congestion prevention. In admission control, after congestion threat has been signalled, no more connections are allowed to be set up until the congestion has gone away. Admission control is crude but simple and robust to implement, and has been used in telephone systems for decades.

5.4 Flow control

Problems of congestion control, like congestion collapse, are largely related to the flow control of TCP (Transmission Control Protocol). TCP adjusts a source node's transmission rate according to the rejected number of datagrams (TCP considers it as a congestion measure) in the network. During the flow control of TCP session, a sender transmits W (W=size of the transmission window) datagrams per time unit and starts to wait for acknowledgements from the receiver. The receiver sends an acknowledgement signal for each datagram, which it has received. If all the datagrams are received, the source increases the size of the window (additive increment), while if a datagram is dropped the size of W is halved (multiplicative decrement). This is also called a *sliding-window* scheme. The drawback of it is that the transmission rate is decreased only after the detection of datagrams losses, which causes a time delay (due to round trip time, RTT) and results in buffer overflows in routers and further losses of datagrams. Hence, it is obvious that the flow control of TCP with the sliding window scheme is not sufficient for flow and congestion control in terms of the network performance and overall quality of service.

On the other side, real-time flows with stringent delay requirements make use of UDP (User Datagram Protocol), which lacks the mechanism to regulate the amount of data being transmitted. UDP does not return acknowledgements and cannot signal congestion to the sender. The inability of UDP flows to regulate transmission rate at the transport layer makes them especially vulnerable to congestion. Therefore, for the UDP sessions, applications have to provide some form of flow control on their own.

6. Congestion and flow control in WLANs

In access networks, like WLANs, congestion occurs when the load on the network is temporarily greater than the resources. Congestion typically causes packet loss due to collisions, which arises when several nodes try to send at the same time, *i.e.*, try to do channel reservation at the same time with CSMA/CA MAC, decreasing significantly transmission rate and increasing dramatically delay.

In WLANs delay and throughput are very much dependent on the packet size, packet transmission interval, and the node connection density. Therefore, in a congested state one

can either decrease load by denying and/or degrading services or reduce channel access competition by access control and/or packet size and transmission interval control.

Congestion can be identified via monitoring, *e.g.*, the *percentage share of discarded datagrams*, *average queue lengths*, and the *percentage share of datagrams that are timed out and retransmitted* on access points, and monitoring the *average value* and *variance of a datagram's delay* on destination nodes. A natural step after monitoring and identification is to transfer information from the congested places (destination nodes, access points) to places where control actions can be performed (source nodes, access points). However, the nodes do not know whether the cause of the packet loss is due to congestion or low signal to noise ratio.

Here we use an embedded fuzzy expert system on the destination nodes to keep WLAN network operating within its capacity. In our system the destination node monitors congestion by measuring *average one-way delay error* and the *change of one-way delay error* (error = delay - target value) as congestion information, defines packet size decrement/increment according to them, and delivers packet size information to the source node.

7. Proportional-integral-derivative controller

A proportional-integral-derivative (PID) control is a widely used feedback control mechanism. A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint and attempts to minimize the error by adjusting the process control inputs. The *proportional* value determines the controller's reaction to the current error, the *integral* value determines the reaction based on the sum of recent errors, and the *derivative* value defines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process, such as the packet payload size of the transmitter, via a control element.

In the developed PID controller, one-way delay error (E_d = proportional value = delay - target value), sum of the recent errors (I_d = integral value), and the change of error (ΔE_d = derivative value) are used as the input values. The output value of the controller is the change of the packet payload size. The new packet payload size is the change of the packet payload size + earlier packet size. The developed controller can be presented in the equation form as follows:

$$P_i(t) = K_p \times E_d(t) + K_i \times \int_{-3}^{0} E_d(t)\, dt + K_d \times \frac{\Delta E_d(t)}{dt}, \qquad (1)$$

where P_i is the change of the packet payload size, K_p (=0.75) is a proportional amplifier, K_i (=0.20) is an integration amplifier, K_d (=0.1) is a derivation amplifier, and t is time.

The controller is located at the user terminal. The controller was designed to update the transmission packet size on the source in order to reach an application dependent target end-to-end delay with the maximum throughput in the prevailing channel conditions. For example in VoIP calls (Andrews et al., 2007) and in action games (Balakrishnan & Sadasivan, 2007), it is preferred that the absolute one-way delay should remain below 100 ms. Maximum throughput instead of the fixed minimum required throughput is needed for example for the video conversations with scalable video coding. Video conversations have a strict end-to-end delay requirement but flexible throughput requirement. Therefore, with the same delay but higher throughput it is possible to use better video coding for higher quality of videos.

8. Fuzzy flow controller

Fuzzy set theory was originally presented by L. Zadeh in his seminal paper "Fuzzy Sets" in *Information and Control 1965* (Zadeh, 1965). Fuzzy logic was developed later from fuzzy set

theory primary to reason with uncertain and vague information and secondary to represent knowledge in operationally powerful form. In the fuzzy set theory the name *fuzzy sets* are used to distinguish them from the *crisp sets* of the conventional set theory. The characteristic function of a crisp set C, $\mu_C(u)$, assigns a discrete value (usually either 0 or 1) to each element u in the universal set U, *i.e.*, it discriminates members and non-members of the crisp set (then for each element u of U, either $u \in C$ or $u \notin C$). The characteristic function can be generalized in fuzzy set theory so that the values assigned to the elements u of the universal set U fall within a prespesified range (usually to the unit interval [0, 1]) indicating the membership grade of these elements in the fuzzy set F. Then it is not necessary that either $u \in F$ or $u \notin F$. The generalized function is called *membership function* and the set defined with the aid of it is a *fuzzy set*, respectively. The membership function assigns to each $u \in U$ a value from the unit interval [0, 1] instead of dual value set {0,1}.

A fuzzy control was originally developed to include a human operator's or system engineer's expertise, which does not lend itself to being easily expressed in PID -parameters or differential equations but rather in situation/action rules. In this study a fuzzy expert system based controller was developed to handle the problems of large overshoot, large steady state error and long-rise time that are evident in the classical systems (Chang & May, 1996). Li & Lau (1989) have shown that the fuzzy proportional-integral controller is less sensitive to large parametric changes in the process and is comparable in performance to the conventional PI controller for small parametric changes. In the fuzzy control system the input and output variables are represented in linguistic form after fuzzyfication of physical values into linguistic form. In this application the input variables are the *average one-way delay error* and the *change of one-way delay error*, the output value is the *packet size increment*. This is so called *two-input, single output control strategy*. For the accurate one-way delay measurement, the clocks of the network nodes were synchronized by beacon signals broadcasted every 100 ms from the access point.

The major components of an expert system are the knowledge base and inference engine. The knowledge base contains the expert-level information necessary to solve domain specific problems, *i.e.*, the knowledge bases are domain specific and nontrasferable. The information is generally presented in the rule form, although, *e.g.*, semantic nets and belief networks are also used. The inference engine interacts both with the knowledge base and a system memory, which includes the facts about the current problem. Pattern matching occurs between the rules in the knowledge base and the recorded facts in the working memory to select the relevant rules applicable (Leondes, 1998).

In fuzzy expert system based control applications, a rule base includes a control policy, which is usually presented with linguistic conditional statements, *i.e.*, if-then rules. Here we present the rule base in the matrix form and the reasoning is done by linguistic equations, see Juuso (1992) and Frantti & Mahonen (2001). Linguistic equations provide a method for developing and tuning adaptive expert systems without rule-based programming. The main advantages of the linguistic equations are the compact size of rule base and computational efficiency. Linguistic equations are also effective in presentation and solving massive rule bases which easily lead to maintenance and testing problems. In the inference engine, the control strategy produces the linguistic control output, which is transformed back into the physical domain in order to find the crisp control output value for the packet size increment. In fuzzy set theory reasoning can be done either using *composition based* or *individual based inference*. In the former all rules are combined into an explicit relation and then fired with fuzzy input whereas in the latter rules are individually fired with crisp input and then combined into one overall fuzzy set. Here we used individual based inference with Mamdani's implication. The main reason for the choice was its easier implementation (the results are equivalent for both

methods when Mamdani's implication is used). Interested reader finds more information about fuzzy controllers, for example, from (Driankov et al., 1994).

8.1 Fuzzy expert system

In the developed fuzzy expert system (FES) based controller, the fuzzy proportional, integral, and derivative parts (FPID) are included to improve the controller's performance. The structure of the developed fuzzy controller for the packet size definition is presented in Figure 1. The fuzzy controller monitors incoming traffic, defines the change of packet size for the source node, and transmits a packet size control command to the source node by acknowledgements. The actual fuzzy system, which is located at the user terminal, has three modules: a fuzzyfication module, a reasoning module and a defuzzyfication module.
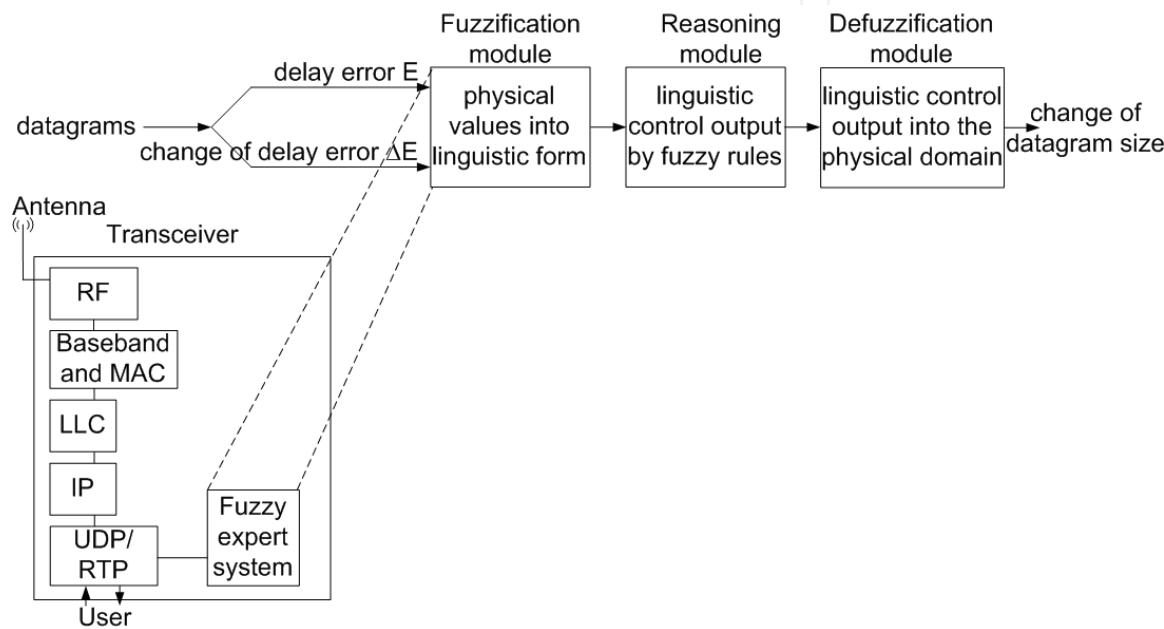

Fig. 1. Fuzzy model for packet size control.

The one-way delay error (E_d) and the change of it (change of the delay error, ΔE_d) are used as the input values for the fuzzy reasoning model. Input variables are represented in linguistic form after fuzzyfication in the fuzzyfication module. Fuzzyfication procedure is illustrated in Figures 2 and 3. In Figure 2, the delay error E_d is -24.92 ms, which is *negative big* at the grade of 0.48 and *negative small* at the grade of 0.52. The change of delay ΔE_d is 6.46 ms, which from one's part is *zero* at the grade of 0.77 and from other part is *positive small* at the grade of 0.23, see Figure 3.

In this application, a linguistic model of a system was described by linguistic relations. The linguistic relations form a rule base (25 rules, see Figure 5) that can be converted into numerical equations. Suppose, as an example, that X_{ij}, $i=1,2$; $j = 1,..., m$ (j is uneven number), is a linguistic level (*e.g.,negative big, negative small, zero, positive small, and positive big*) for a variable X_i. The linguistic levels are replaced by integers $\frac{-(j-1)}{2},...,-2,-1,0,1,2,...,\frac{(j-1)}{2}$. The direction of the interaction between fuzzy sets is presented by coefficients $A_{ij}=\{-1,0,1\}$, $i=1,2$; $j = 1,..., m$. This means that the directions of the changes in the output variable decrease or increase depending on the directions of the changes in the input variables (Juuso, 1993). Thus a compact equation for the output Z_{ij} is:

$$\sum_{j=1}^{m}\sum_{i=1}^{2} A_{ij}X_{ij} = Z_{i,j}. \tag{2}$$

The mapping of linguistic relations to linguistic equations for this application is described in Figure 5. For example, we can read from Figure 5 that *IF E_d IS negative small AND ΔE_d IS zero THEN the change of packet size IS positive small*. In linguistic equations this can be presented as $\lceil \frac{(-1*-1+-1*0)}{2} \rceil = 1$. A more detail reasoning example is given in Section 8.2.

The most important properties for a set of rules are *completeness, consistency, continuity* and *interaction*. *Completeness* of rules means that all kinds of combinations of input variables results in an appropriate output value. The rule base is *consistent* if it does not contain any contradiction[1]. It can be formulated as in (Driankov et al., 1994): A set of rules is inconsistent if there are at least two rules with the same rule-antecedent and different rule-consequent. *Continuity* means that neighboring rules have no output fuzzy sets with an empty intersection. Definitions of neighboring rules are given for example in (Driankov et al., 1994) as follows: two rules are neighbors, if their cells are neighbors in matrix representation of a rule base. An *interaction* of a set of rules is defined many ways in the literature. Driankov et al. (1994) state that a set of fuzzy rules interacts if composition based inference does not equal individual based inference.

In the defuzzyfication module the control strategy produces the linguistic control output, which is transformed back into the physical domain to find the crisp output value for the change of packet size. In the defuzzyfication phase the center of area method (CoA) was used. The defuzzyfication procedure is illustrated in Figure 4. From Figure 4 it can be seen that the change of packet size is *positive small* at the grade of 0.52 and *positive big* at the grade of 0.48. The crisp output value is the center of the area, *i.e.*, the new packet size is 43 bits bigger than the earlier value.

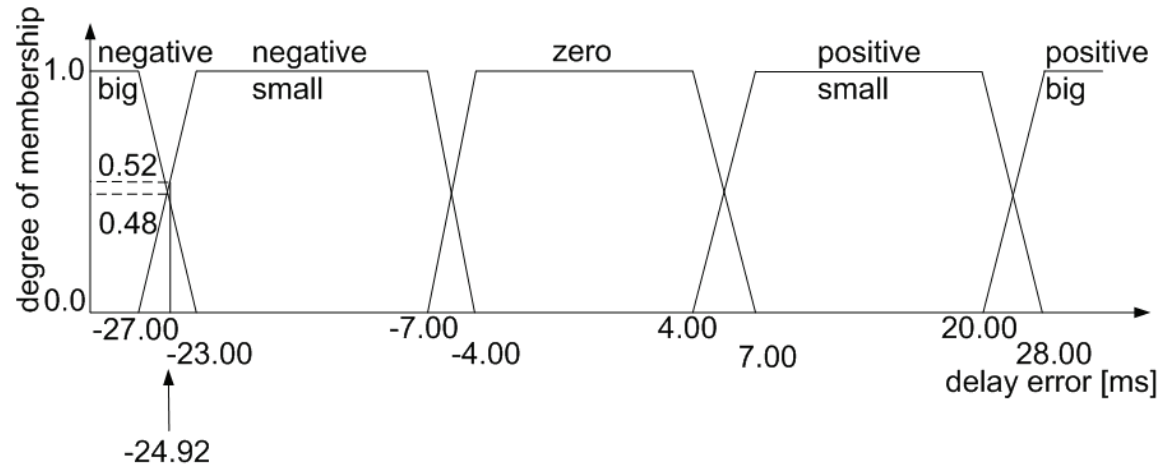Fig. 2. Fuzzy membership functions for the E_d.

8.2 Reasoning example

The developed fuzzy expert system was designed to update the transmission packet size in order to reach a target end-to-end delay with the maximum throughput in the prevailing channel conditions. Consider as an example, that the E_d is -24.92 ms, which is after

---

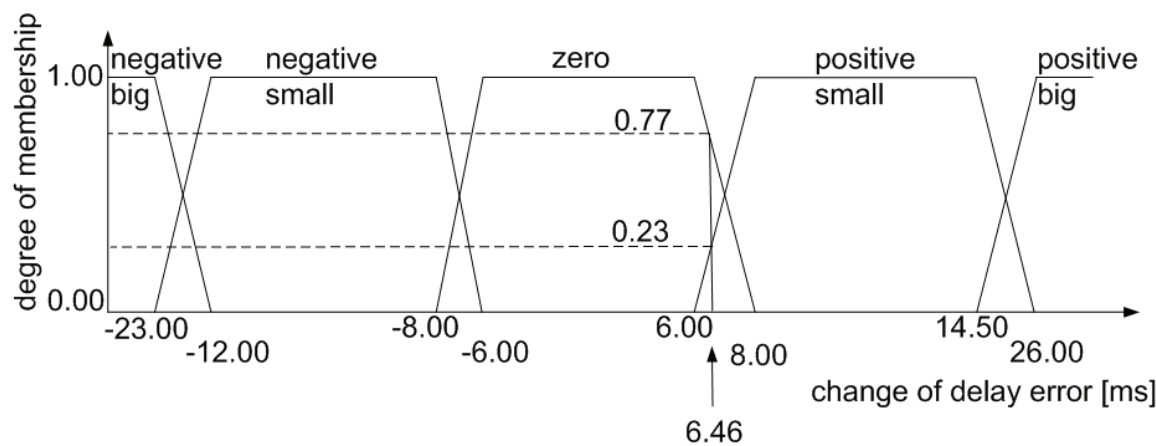[1] In the literature it is also defined like continuity below.

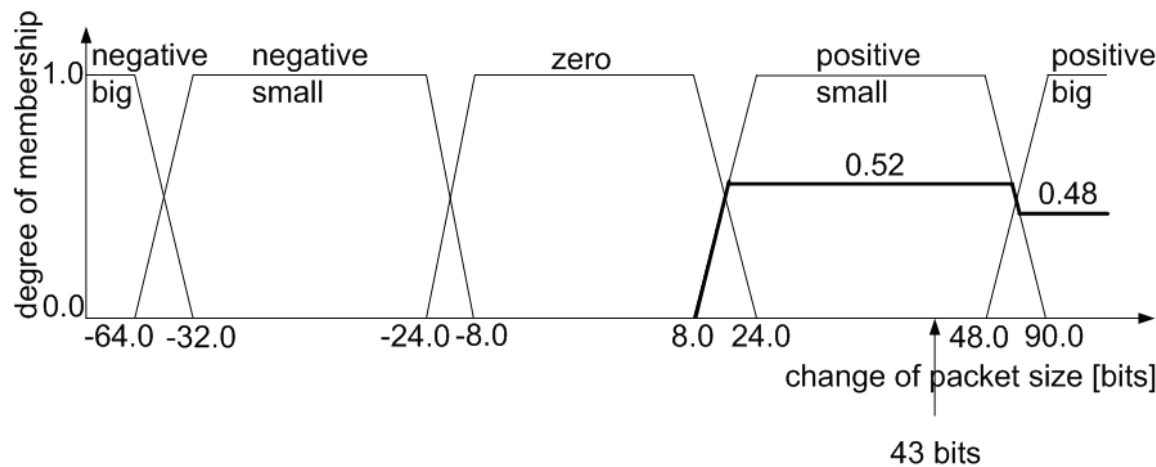Fig. 3. Fuzzy membership functions for the ΔE_d.

Fig. 4. Fuzzy membership functions for the change of packet size.

fuzzyfication in linguistic form *negative big* at the grade of membership 0.48 and *negative small* at the grade of membership 0.52 (see Figure 2). Suppose that the ΔE_d is 6.46 ms, which is after fuzzyfication in linguistic form *zero* at the grade of membership 0.77 and *positive small* at the grade of membership 0.23 (see Figure 3). Now we can read from Figures 2, 3 and 5 that

IF E_d IS NB at the grade 0.48 AND ΔE_d IS ZE at the grade 0.77 THEN the change of packet size IS PB at the grade 0.48

IF E_d IS NB at the grade 0.48 AND ΔE_d IS PS at the grade 0.23 THEN the change of packet size IS PB at the grade 0.23
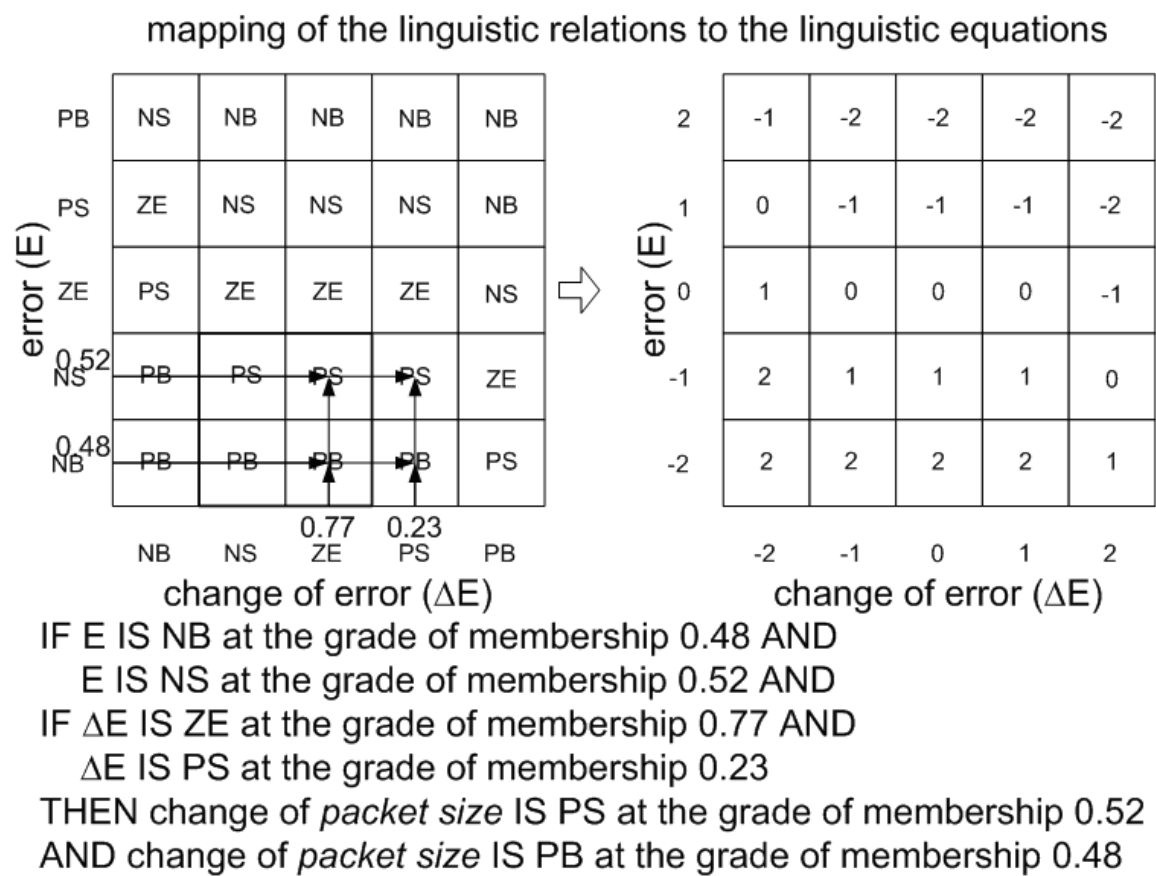
IF E_d IS NS at the grade 0.52 AND ΔE_d IS ZE at the grade 0.77 THEN the change of packet size IS PS at the grade 0.52

IF E_d IS NS at the grade 0.52 AND ΔE_d IS PS at the grade 0.23 THEN the change of packet size IS PS at the grade 0.23

In linguistic equations this can be presented as follows:

$\lceil \frac{(-2*-2+-1*0)}{2} \rceil = 2$ at the grade min(0.48,0.77)

mapping of the linguistic relations to the linguistic equations

Fig. 5. Fuzzy rule base and mapping of the linguistic relations to the linguistic equations.

$$\lceil \frac{(-2*-2+-1*1)}{2} \rceil = 2 \text{ at the grade min}(0.48,0.23)$$

$$\lceil \frac{(-2*-1+-1*0)}{2} \rceil = 1 \text{ at the grade min}(0.52,0.77)$$

$$\lceil \frac{(-2*-1+-1*1)}{2} \rceil = 1 \text{ at the grade min}(0.52,0.23)$$

where $\lceil \rceil$ returns the next highest integer value by rounding up the value if necessary. Using individual based inference with Mamdani's implication the weight value is *positive big* at the grade of membership 0.48 (max(0.48,0.23)) and *positive small* at the grade of membership 0.52 (max(0.52,0.23)). Therefore, the crisp output value is 43 bits (see Figure 4), which is used in the user equipment to update a new packet size to be 43 bits bigger than earlier. The rule base does not allow the packet size decrease below 256 bits or increase over 11520 bits but keeps the packet size between [256, 11520] bits.

8.3 Adaptation

In order to adapt to application dependent delay requirements, the developed fuzzy expert system needs as an input an application dependent target (maximum acceptable) delay value. It then controls real-time traffic flow(s) by optimizing packet sizes for the target delay on the prevailing channel conditions. In addition, to keep membership functions and inference logic independent from the absolute dependency of delay value and packet size, the expert system use relative input values (*delay error* and the *change of delay error*). The expert system also defines the increment of the packet size as an output value instead of the absolute packet size.

With absolute input variable, like *delay*, the membership functions should be redefined for all the possible target delay values.

8.4 Computational complexity

Flow control by the definition of the packet size on the mobile nodes increases node's computational complexity. The implementation decision of the fuzzy control method is a trade-off between complexity, required computational time, required RAM (Random Access Memory) and program memory and achieved advantages of the algorithm. The fuzzy feedback control also lightly increases communicational load by transmitting application level acknowledgements after every 200 received packets. However, for the UDP sessions, applications have to anyway provide some form of flow control on their own. Fuzzyfication phase requires at the most two $\times$ nine comparisons, two $\times$ two addition and two $\times$ one multiplications. In the comparisons crisp input values are compared to the parts of the membership functions, which cover the dynamic ranges of the input variables, see Figure 2. After the comparisons, when the fired fuzzy label(s) are identified, the degree of membership is defined by multiplying the interval of corner point and crisp value by the angle of the line. Multiplication is not needed if the crisp point sets to the top area, *i.e.*, the degree of membership is 1.0. Reasoning process for linguistic equations requires in the worst case eight multiplications, four additions, four divisions, and six min/max comparisons. In the defuzzyfication phase, it is required at most two $\times$ two additions and two $\times$ five multiplications for the definition of horizontal component of the center of area. All in all, the developed control method increases at most 56 computations for fuzzy packet size definition. According to Koomey (2010) it can be estimated that one operation requires 1.2-1.8 nJ and thus 56 operations requires about 84 nJ, if the value 1.5 nJ/operation is used. The estimated energy consumption per transmitted packet is then $\frac{84}{200}$ nJ = 0.42 nJ, which is less than or equal to $\frac{1}{3}$ of the energy consumption one elementary operation requires.

Parameter	Value
Simulation time	200 s
Wireless hosts	6-10
Protocols	IP/UDP/RTP
MAC	CSMA/CA
MAC data rate	11 Mbit/s
carrier frequency	2.4 GHz
transmitter power	2.0 mW
thermal noise	-110 dBm
sensitivity	-85 dBm
snirThreshold	4 dB
Simulation area	600 x 400 m
VoIP data rate	64 kbit/s
Video Phone data rate	384 kbit/s
Interactive game data rate	40 kbit/s

Table 1. Parameters for the simulations.

9. Network simulations

The simulation studies were done with OMNeT++ 4.0 simulator (http://www.omnetpp.org) with the INETMANET framework. The simulation model consisted of 6-10 wireless hosts in an infrastructure, *i.e.* basic service set (BSS), mode and one IEEE 802.11*b* WLAN access point. The nodes were close to each other (max. distance between any two nodes was about 12.7 m). The distance from a host to the access point (AP) varied between 14.1 and 26.9 m. The nodes were not moving and it was assumed that the nodes were synchronized using, *e.g.*, access point's beacon message. The most important simulation parameters are shown in Table 1.

In *Scenario 1*, the performance of VoIP traffic was studied. The VoIP traffic data rate was 64 kbit/s. The used fixed packet size was 400 bits, *i.e.*, the packet interval was 6.25 ms. The VoIP calls were made in pairs, *i.e.*, host0 and host1 formed one pair, host2 and host3 another pair, and so on. All hosts measured the delay for the packets, used the developed packet size optimization algorithms to calculate the optimum packet size for 25 ms target delay (for protocol, queueing and propagation) and reported it to its pair after every 200 packets by sending an UDP acknowledgement message. Then the pair adjusted its packet size. Also the packet interval was adjusted in order to keep the data rate constant.

Scenario 2 included in addition to VoIP traffic also other real-time applications. Video phone application had 384 kbit/s data rate with 7680 bits packet size and 20 ms packet interval. Interactive game had 40 kbit/s data rate with 400 bits packet size and 10 ms packet interval. Also these applications were used in pairs, too. All hosts measured the delay for the packets, used the developed packet size optimization algorithms to calculate the optimum packet size for 25 ms target delay, and reported it back to its peer after every 200 packets by an UDP acknowledgement message.

10. Results

10.1 Delay and throughut

The developed flow controllers were designed for interactive real-time application such as VoIP calls, video calls, and interactive games to reach an application dependent target end-to-end delay and to maximize the number of real-time connections in an access point's coverage area. Therefore, the conducted simulation scenarios measured delay and throughput when the packet payload size was fixed, adjusted by the developed PID controller, and adjusted by the developed fuzzy controller.

The delay and throughput evaluations were performed as a function of packet size and varying number of connections. The results in our earlier publication (Frantti & Majanen, 2010) showed that there is an optimal packet size with respect to the overall delay and packet loss rate, which depends on the number and type of real-time connections. In practise, the amount of background traffic changes as a function of time and it is not possible to manually choose optimal fixed packet sizes for the current background traffic level. Therefore, we ended to use the value of 400 bits because as an average it seems to give the best results.

The overall delay in VoIP and video calls contains delays in the MAC layer, the link layer, the TCP/IP protocol stack, propagation delay in the radio channel, queueing delays in intermediate nodes, speech and video coding delays, jitter buffer delay, and the lookahead delay of codec. The size of the jitter buffer was varied from 70 ms to 110 ms depending on the packet payload sizes. For longer packets, the jitter buffer was shorter than for shorter packets due to the longer delay caused by speech coding for longer packets in order to keep the overall delay for all the packets below 150 ms. According to (Andrews et al., 2007), absolute delay should not exceed 150 ms for good voice communication quality and it is preferred

VoIP traffic	Delay			Throughput		
	OF [ms]	FC [ms]	PID [ms]	OF [Kbit/s]	FC [Kbit/s]	PID [Kbit/s]
Host one	27.6	2.1	1.0	30.5	64.8	64.2
Host two	28.0	2.0	4.9	29.3	63.2	62.6
Host three	27.7	2.0	6.1	32.2	64.0	63.4
Host four	28.2	2.1	5.6	29.6	64.0	63.4
Host five	28.0	1.9	5.7	29.5	63.9	63.3
Host six	27.7	1.9	5.6	30.5	64.1	63.5

Table 2. Delay and throughputs in *Scenario 1* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limit 64 Kbit/s. VoIP traffic. Six interactive calls.

VoIP traffic	Delay			Throughput		
	OF [ms]	FC [ms]	PID [ms]	OF [Kbit/s]	FC [Kbit/s]	PID [Kbit/s]
Host one	64.0	3.8	17.7	8.8	63.8	57.9
Host two	62.2	5.6	17.4	9.0	62.3	56.5
Host three	73.8	5.6	22.8	12.7	63.0	57.4
Host four	67.4	5.8	20.2	11.7	63.0	57.2
Host five	95.4	3.8	25.4	13.2	62.8	57.3
Host six	86.5	4.0	19.9	13.5	63.2	57.7
Host seven	123.0	5.4	31.4	11.1	63.2	57.3
Host eight	113.9	5.8	30.1	12.3	62.8	57.3

Table 3. Delay and throughputs in *Scenario 1* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limit 64 Kbit/s. VoIP traffic. Eight interactive calls.

that the absolute delay should remain below 100 ms for VoIP calls. Therefore, the delay from the jitter buffer and speech coding was designed to be a constant 125 ms, which consist of 70-110 ms delay of jitter buffer, 10-50 ms delay of audio samples coding and 5 ms lookahead delay of speech coding algorithm. Due to constant 125 ms delay, the protocol, queueing and propagation delay should remain below 25 ms to fulfil 150 ms delay limit.

Table 2 presents delay (protocol + queueing + propagation delays) and throughput for the fixed packet size of 400 bits, for the fuzzy controlled flows, and for the PID controlled flows with three connection pairs, *i.e.*, six hosts. The average delays were 27.9 ms for the fixed packet size, 2.0 ms for the fuzzy controlled flows, and 4.8 ms for the PID controlled flows. The average throughputs were 30.3 Kbit/s for the fixed packet size, 64.0 Kbit/s for the fuzzy controlled flows, and 63.4 Kbit/s for the PID controlled flows.

Table 3 presents delay (protocol + queueing + propagation delays) and throughput for the fixed packet size of 400 bits, for the fuzzy controlled flows, and for the PID controlled flows with four connection pairs. The average delays were 85.8 ms for the fixed packet size, 5.0 ms for the fuzzy controlled flows, and 23.1 ms for the PID controlled flows. The average throughputs were 11.5 Kbit/s for the fixed packet size, 63.0 Kbit/s for the fuzzy controlled flows, and 57.3 Kbit/s for the PID controlled flows.

VoIP traffic	Delay			Throughput		
	OF [ms]	FC [ms]	PID [ms]	OF [Kbit/s]	FC [Kbit/s]	PID [Kbit/s]
Host one	77.1	8.6	0.9	8.1	59.5	15.8
Host two	75.3	11.1	1.2	6.3	55.8	15.1
Host three	114.7	17.2	9.4	7.3	56.5	14.9
Host four	98.8	15.9	67.8	7.8	56.4	15.4
Host five	139.9	9.2	0.9	8.3	56.7	18.1
Host six	116.8	9.3	85.6	10.0	56.8	16.1
Host seven	160.7	23.1	104.5	8.8	57.1	18.3
Host eight	151.2	21.7	103.8	8.4	56.7	17.5
Host nine	182.1	24.8	133.4	7.0	55.8	12.6
Host ten	150.5	10.5	107.8	9.1	58.0	16.7

Table 4. Delay and throughputs in *Scenario 1* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limit 64 Kbit/s. VoIP traffic. Ten interactive calls.

Table 4 presents delay (protocol + queueing + propagation delays) and throughput for the fixed packet size of 400 bits, for the fuzzy controlled flows, and for the PID controlled flows with five connection pairs. The average delays were 126.7 ms for the fized packet size, 15.1 ms for the fuzzy controlled flows, and 61.5 ms for the PID controlled flows. The average throughputs were 8.1 Kbit/s for the fized packet size, 56.9 Kbit/s for the fuzzy controlled flows, and 16.1 Kbit/s for the PID controlled flows.

Table 5 presents delay (protocol + queueing + propagation delays) and throughput in *Scenario 2* for the fixed packet size of 400 bits, for the fuzzy controlled flows, and for the PID controlled flows with VoIP (throughput requirement 64 Kbit/s) call, video call (throughput requirement 384 Kbit/s) and interactive game (throughput requirement 40 Kbit/s) connection pairs. With the developed fuzzy and PID controllers delay and throughput limits are perfectly achieved. The applications work relatively well also with the fixed 400 bits packet size except of a bit lower throughput than required for video call and interactive game. Therefore, it can be stated from the results that controllers are not necessarily required with the very low network load if an optimal fixed packet size is known. However, in practise, the amount of background traffic changes as a function of time and it is not possible to manually choose optimal fixed packet sizes for the current connection, which further enhances the need of the control even for the low network load.

Table 6 presents delay (protocol + queueing + propagation delays) and throughput in *Scenario 2* for the fixed packet size of 400 bits, for the fuzzy controlled flows, and for the PID controlled flows with two VoIP (throughput requirement 64 Kbit/s) calls, two video calls (throughput requirement 384 Kbit/s) and one interactive game (throughput requirement 40 Kbit/s) connection pairs. It can be seen that the developed controllers can fulfil the delay time requirement. The fuzzy controller responds satisfactorily to the throughput requirements even if the throughput is a bit too low for video calls and interactive game connection.

Table 7 presents delay and thropughput of two pairs of VoIP calls, video calls, and interactive games. It can be seen that only the fuzzy controller manage to keep delay within the requirement. The throughput is a bit lower than required for perfect connection but still very near the perfect level.

Connection types	Delay			Throughput		
	OF [ms]	FC [ms]	PID [ms]	OF [Kbit/s]	FC [Kbit/s]	PID [Kbit/s]
VoIP connection	12.4	2.6	1.0	62.3	63.6	64.0
Video call	12.9	4.4	3.4	375.4	384.0	384.0
Interactive game	12.1	2.5	2.6	38.8	40.0	40.0

Table 5. Delay and throughputs in *Scenario 2* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limits were 64 Kbit/s for VoIP call, 384 Kbit/s for video call, and 40 Kbit/s for interactive game. One VoIP call, one video call, and one interactive game connection.

Connection types	Delay			Throughput		
	OF [ms]	FC [ms]	PID [ms]	OF [Kbit/s]	FC [Kbit/s]	PID [Kbit/s]
VoIP connection 1	100.6	10.2	30.4	9.7	61.8	53.7
VoIP connection 2	127.9	13.0	34.5	9.6	61.9	52.8
Video call 1	82.2	9.2	14.3	79.3	371.8	325.1
Video call 2	87.4	10.2	15.4	79.2	371.5	324.1
Interactive game 1	141.5	10.3	31.4	7.9	38.7	33.7

Table 6. Delay and throughputs in *Scenario 2* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limits were 64 Kbit/s for VoIP call, 384 Kbit/s for video call, and 40 Kbit/s for interactive game. Two VoIP call, two video call, and one interactive game connection.

The aim of the developed fuzzy flow management system is to adjusts appropriate packet size and transmission interval of the source node's constant bit rate real-time traffic flows for prevailing network conditions to achieve application dependent quality of service requirements with the optimal network capacity. From the results (see Tables 2 - 7) it can be seen that with the increasing load, the delay increases and throughput degrades smoothly towards QoS limits when the fuzzy controller was used. Hence, in order to guarantee the quality of service of the different applications, an admission control is required either to accept or to deny new connections depending on the prevailing network conditions. In our case the access point, with the fuzzy controller in network nodes, allows new prioritized real-time connections when the required overall capacity of them remains below 900 Kbit/s. For the PID controller, the capacity limit should be around the 480 Kbit/s.

10.2 Response times
Table 8 presents throughput and corresponding averaged packet sizes of real-time traffic with different amount of background traffic for 100 ms target delay. The optimum packet size value

Connection	Delay			Throughput		
	OF	FC	PID	OF	FC	PID
types	[ms]	[ms]	[ms]	[Kbit/s]	[Kbit/s]	[Kbit/s]
VoIP connection 1	120.4	18.0	112.2	7.4	59.1	7.6
VoIP connection 2	144.2	14.7	127.6	7.8	60.1	7.4
Video call 1	87.0	14.8	90.5	63.6	354.0	61.1
Video call 2	92.9	15.4	82.7	72.7	357.1	75.8
Interactive game 1	173.0	16.2	161.0	6.9	37.4	5.9
Interactive game 2	236.0	12.7	175.9	4.8	37.5	5.9

Table 7. Delay and throughputs in *Scenario 2* when the fixed packet size of 400 bits, fuzzy controller (FC) and PID controller (PID) were used. Protocol, queueing and propagation delay limit was 25 ms. Throughput limits were 64 Kbit/s for VoIP call, 384 Kbit/s for video call, and 40 Kbit/s for interactive game. Two VoIP call, two video call, and two interactive game connection.

Background	Average packet size			Throughput		
	OF	FES	PID	OF	FES	PID
traffic	[bits]	[bits]	[bits]	[Kbit/s]	[Kbit/s]	[Kbit/s]
(0.010,0.100)	10900	10552	10538	1518	2149	2130
(0.010,0.090)	9750	9366	9324	1285	1907	1874
(0.010,0.085)	8800	8897	8455	1180	1804	1763
(0.010,0.080)	8200	8276	7876	1051	1686	1593
(0.010,0.075)	7100	7103	7034	900	1457	1429
(0.010,0.070)	6200	6207	5793	715	1255	1175

Table 8. Throughputs and corresponding averaged packet sizes with different amount of background traffic. Delay limit 100 ms. Initial packet size was 256 bits. One-directional traffic. OF = Optimized fixed.

depends on the amount of traffic on the network. It was defined separately for all background traffic levels in each scenarios by measuring and depicting delay as a function of packet size by a large set of simulation runs. Therefore, it is an optimal value for the observed unchanged circumstances. For the results shown in Table 8, the transmitting host had an application that sent a packet to another host every 1 ms starting with the packet size of 256 bits. Receiving host measured the delay for the packets, used the developed packet size optimization algorithms to calculate the optimum packet size for 100 ms target delay, and reported it to the transmitting host after every 200 packets by sending an acknowledgement message. The surrounding nodes transmit packets at random intervals i, where $i \in [0.010\,s, 0.070\,s]$ - $i \in [0.010\,s, 0.100\,s]$. The conducted simulations measured also rise and settling times of the controllers with the different level of the background traffic. For example, Figures 6 - 7 a and b present throughput as a function of time in, when the packet size was adjusted by the FES and PID controllers

(a)   (b)
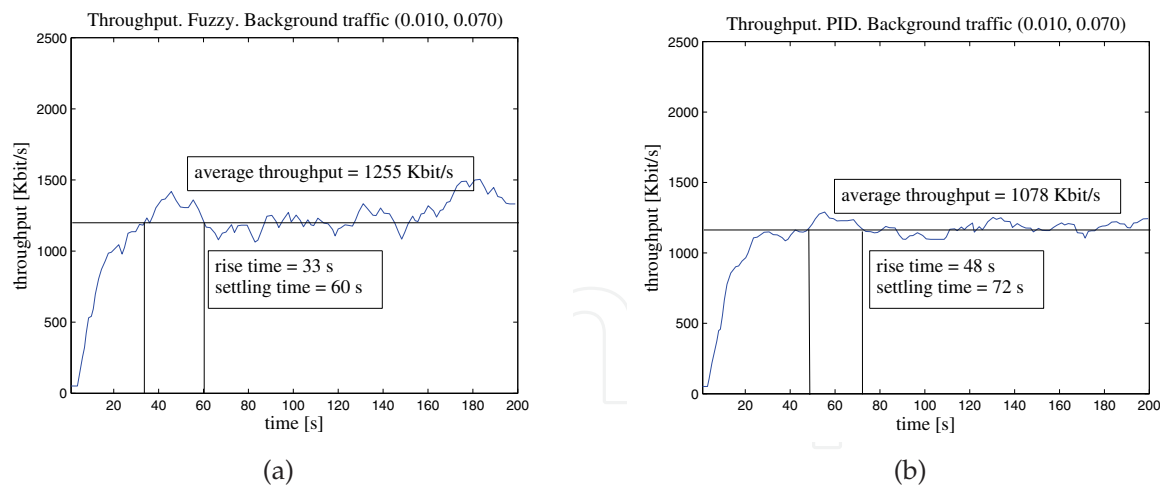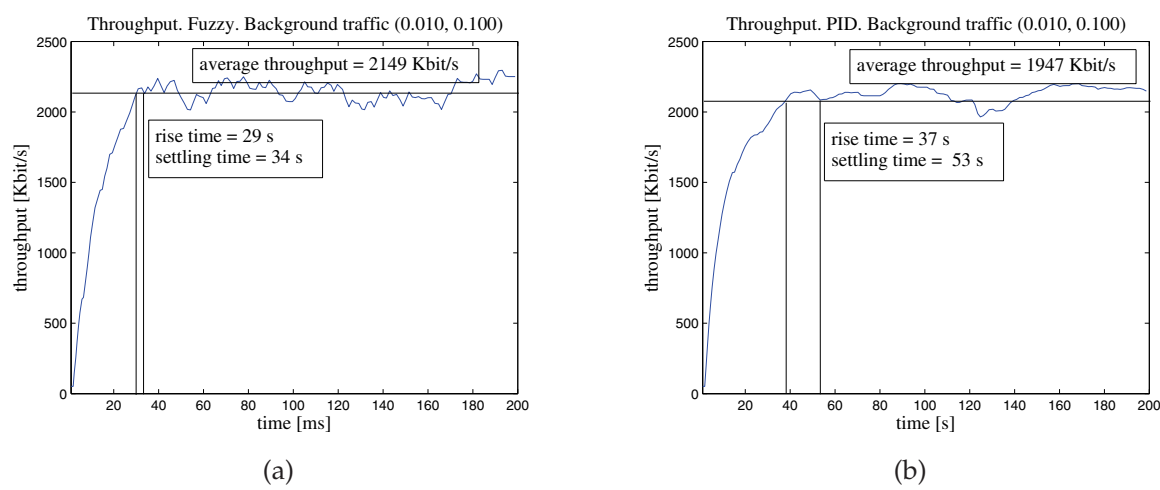
Fig. 6. Throughput as a function of time when the packet size was adjusted by a.) the FES and b.) the PID controller and the surrounding nodes transmit packets at random intervals i, where $i \in [0.010\,s, 0.070\,s]$.

(a)   (b)

Fig. 7. Throughput as a function of time when the packet size was adjusted by a.) the FES and b.) the PID controller and the surrounding nodes transmit packets at random intervals i, where $i \in [0.010\,s, 0.100\,s]$.

and the target delay was 100 ms. The packet transmission interval of surrounding nodes was varied in Figures 6 - 7 from $i \in [0.010\,s, 0.070\,s]$ to $i \in [0.010\,s, 0.100\,s]$.

Table 9 presents rise and settling times of the controllers with 100 ms target delay when the packet transmission interval of surrounding nodes is varied from $i \in [0.010\,s, 0.070\,s]$ to $i \in [0.010\,s, 0.100\,s]$. The average (averaged over the different amount of disturbing background traffic of surrounding nodes) rise and settling times were 41.5 s and 53.2 s for the FES based controller, and 58.5 s and 78.3 s for the PID controller. The developed controllers manage to set packet payload size values to the prevailing optimum level very fast and accurately. However, the rise and settling times of the FES are about 29 % and 32 % lower than for the PID, *i.e.*, it can be stated that the FES controller adapts faster and adjust better to traffic load changes, which is an important feature especially in congestion situation.

Background traffic	Rise time		Settling time	
	FES [s]	PID [s]	FES [s]	PID [s]
(0.01,0.100)	29	37	34	53
(0.01,0.090)	47	88	58	123
(0.01,0.085)	42	62	60	71
(0.01,0.080)	38	53	44	72
(0.01,0.075)	60	63	63	79
(0.01,0.070)	33	48	60	72

Table 9. Rise and settling times of controllers with different amount of background traffic. Initial packet size was 256 bits. Delay limit was 100 ms.

11. Conclusion

This chapter considered an embedded fuzzy control system for cognitive flow management of delay sensitive real-time traffic to improve quality of service (QoS). The management system adjusts transceivers' traffic flow(s) for prevailing network conditions to achieve application dependent delay and throughput quality of service requirements with the optimal network capacity. The fuzzy flow management system was compared to conventional PID control system. The controllers were located at user terminals. The models were validated by simulating voice over IP (VoIP) calls, video phone conversations, and interactive games in OMNeT++ network simulator.

The results showed that the developed controller manages to set packet payload size values to the prevailing optimum level very fast and accurately and they also managed to keep average delay below the target value. For the VoIP conversations, the fuzzy flow management controller doubles the amount of quality controlled connections compared to fixed 400 bits packet payload size calls and PID controlled calls. Therefore, we can state that the developed model enables WLANs to increase the number of concurrent users and improve quality of the real-time connections. The fuzzy control system also adapt to various application level requirements, like an application dependent delay limit, with low computational complexity.

12. Acknowledgement

13. References

Ahmad, S., Mustafa, A., Ahmad, H., Bano, A. & Hosam, A. (2009). Comparative Study of Congestion Control Techniques in High Speed Networks, *(IJSIS) International Journal of Computer Science and Information Security* 6(2): 222–231.

Andrews, J., Ghosh, A. & Muhamed, R. (2007). *Fundamentals of WiMAX - Understanding Broadband Wireless Networking*, first edn, Prentice Hall, United States.

B. Bing (2002). *Wireless Local Area Networks: The New Wireless Revolution*, 1st edition edn, John Wiley & Sons, Inc., New York.

Bakshi, B., Krishna, P., Vaidya, N. & Pradhan, D. (1997). Improving Performance of TCP over Wireless Networks, *ICDCS '97: Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS '97)*, IEEE Computer Society, Washington, DC, USA, pp. 365–373.

Balakrishnan, M. & Sadasivan, M. (2007). Mobile Interactive Game Interworking in IMS, *White Paper. http://www.infosys.com/offerings/engineering-services/product-engineering/white -papers/Documents/mobile-gaming-paper.pdf* .

Chan, P. M. L., Sheriff, R. E., Hu, Y. F., Conforto, P. & Tocci, C. (2001). Mobility management incorporating fuzzy logic for a heterogeneous IP environment, *IEEE Communications Magazine* 39(12): 42–51.

Chang, P. & May, B. W. (1996). Adaptive Fuzzy Power Control for CDMA Mobile Radio Systems, *IEEE Transactions on Vehicular Technology* 45(2): 225–236.

Chee, K. & David, J. (1989). Packet Data Transmission Over Mobile Radio Channels, *IEEE Trans. on Vehicular Technology* 38: 95–101.

Chien, C., Srivastava, M. B., Jain, R., Lettieri, P., Aggarwal, V. & Sternowski, R. (1999). Adaptive Radio for Multimedia Wireless Links, *IEEE Journal on Selected Area in Communications* 17: 793–813.

Crawford, L. & Marshall, A. (2001). A dynamic and fast packet scheduling algorithm for open and programmable networks, *The Seventeenth UK Teletraffic Theory Symposium*, Dublin, Ireland.

Demers, A., Keshav, S. & Shenker, S. (1989). Analysis and simulation of a fair queueing algorithm, *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, ACM, New York, NY, USA, pp. 1–12.

Driankov, D., Hellendoorn, H. & Reinfark, M. (1994). *An Introduction to Fuzzy Control*, 2nd edition edn, Springer-Verlag, New York.

Floyd, S. & Jacobson, V. (1995). Link-sharing and resource management models for packet networks, *IEEE/ACM Trans. Netw.* 3(4): 365–386.

Frantti, T. & Jutila, M. (2009). Embedded Fuzzy Expert System for Adaptive Weighted Fair Queueing, *Expert Systems with Applications, Elsevier Science* Vol. 36, No. 8: 11390–11397.

Frantti, T. & Mahonen, P. (2001). Fuzzy Logic Based Forecasting Model, *Engineering Applications of Artificial Intelligence* 14(2): 189–201.

Frantti, T. & Majanen, M. (2010). *Internet Traffic Shaping in WLANs by Packet Size Control*, first edn, Nova Science Publishers, Inc., NY.

Frantti, T., Majanen, M. & Sukuvaara, T. (2010). Delay Based Packet Size Control in Wireless Local Area Networks, *ICUFN 2010 The Second International Conference on Ubiquitous and Future Networks, June 16-18, 2010, Jeju Island, Korea. Invited paper.*, Jeju Island, Korea.

Goldsmith, A. & Wicker, S. B. (2002). Design challenges for energy-constrained ad hoc wireless networks, *IEEE Wireless Commun. Mag.* 9(4): 8–27.

Hahne, E. (1986). *Round robin scheduling for fair flow control in data communication networks*, Ph.d. dissertation lids-th-1631, Lab. Inform. Decision Syst., MIT, Cambridge, MA.

Horng, M. F., Lee, W. T., Lee, K. R. & Kuo, Y. H. (2001). An adaptive approach to weighted fair queue with QoS enhanced on IP, *IEEE Region 10 International Conference on Electrical and Electronic Technology*, Vol. 1, pp. 181–186.

Huitema, C. (2000). *Routing in the Internet (2nd ed.)*, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Huusko, J., Vehkapera, J., Amon, P., Lamy-Bergot, C., Panza, G., Peltola, J. & Martini, M. (2007). Cross-layer architecture for scalable video transmission in wireless network, *Signal Processing: Image Communication* 22(3): 317–330.

Juuso, E. (1992). Linguistic Equations Framework for Adaptive Expert Systems, *in* J. Stephenson (ed.), *Modelling and Simulation 1992, Proceedings of the 1992 European Simulation Multiconference*, pp. 99–103.

Juuso, E. K. (1993). Linguistic Simulation in Production Control, *in* R. Pooley & R. Zobel (eds), *UKSS'93 Conference of the United Kingdom Simulation Society*, Keswick, UK, pp. 34–38.

Khoukhi, L. & Cherkaoui, S. (2008). Experimenting with Fuzzy Logic for QoS Management in Mobile Ad Hoc Networks, *IJCSNS International Journal of Computer Science and Network Security* 8(8): 372–386.

Kleinrock, L. (1975). *Queuing Systems*, first edition edn, John Wiley & Sons, New York.

Koomey, J. G. (2010). Outperforming Moore's Law, *IEEE Spectrum* March 2010: 68.

Korhonen, J. & Wang, Y. (2005). Effect of Packet Size on Loss Rate and Delay in Wireless Links, *Proceedings of the 2005 IEEE Conference on Wireless Communications and Networking, vol. 3.*, IEEE Communications Society, New Orleans, LA USA, pp. 1608–1613.

Kumar, S., Raghavan, V. S. & Deng, J. (2006). Medium access control protocols for ad-hoc wireless networks: A survey, *Ad-Hoc Netw. J.* 4(3): 326–358.

Lamy-Bergot, C., Fracchia, R., Mazzotti, M., Moretti, S., Piri, E., Sutinen, T., Zuo, J., Vehkapera, J., Feher, G., Jeney, G., Panza, G. & Amon, P. (2010). Optimisation of multimedia over wireless IP links via X-layer design: an end-to-end transmission chain simulator, *Multimedia Tools and Applications* .

Leondes, C. T. (ed.) (1998). *Fuzzy Logic and Expert Systems Applications*, 6 edn, Academic Press, San Diego, California, USA.

Lettieri, P. & Srivastava, M. B. (1998). Adaptive Frame Length Control for Improve Wireless Link Range and Energy Efficiency, *Proceeding of the IEEE INFOCOM'98*, IEEE communications Society, San Francisco, USA, pp. 564–571.

Leung, K. K., McNair, B., Cimini, L. & Winters, J. H. (2002). Outdoor IEEE 802.11 Cellular Networks: MAC Protocol Design and Performance, *Proc. of the IEEE International Conference on Communications 2002 (ICC 2002), Vol. 1*, pp. 595–599.

Li, T. & Ni, Q. (2005). Performance Analysis of the IEEE 802.11e Block ACK Scheme in a Noisy Channel, *Proc. IEEE BroadNets05*, pp. 551–557.

Li, Y. F. & Lau, F. (1989). Development of Fuzzy Algorithm for Servo Systems, *IEEE Control Systems Magazine* 9(3): 65–72.

Nagle, J. (1985). On packet switches with infinite storage, *RFC 970*, Internet Engineering Task Force.

Nagle, J. (1987). On packet switches with infinite storage, *Communications, IEEE Transactions on [legacy, pre - 1988]* 35(4): 435–438.

Perkins, D. D. & Hughes, H. D. (2002). A survey on QOS support for mobile ad hoc networks, *Wireless Commun. Mobile Comput.* 2: 503–513.

Peterson, L. L. & Davie, B. S. (2007). *Computer networks: a systems approach*, 4th edition edn, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Qu, Q., Pei, Y. & Modestino, J. W. (2005). Cross-layer QoS control for video communications over wireless ad hoc networks, *EURASIP J. Wireless Commun. Network* 5: 743–756.

Sanjay, K. & Hassan, M. (2002). *Engineering Internet Qos*, Artech House, Inc., Norwood, MA, USA.

Sankarasubramaniam, Y., Akyildiz, I. & McLaughlin, S. (2003). Energy Efficiency Based Packet Size Optimization in Wireless Sensor Networks, *Proceedings of the First*

IEEE International Workshop on Sensor Network Protocols and Applications, 2003., IEEE Communications Society, Anchorage, Alaska, USA, pp. 1–8.

Sayenko, A., Hämäläinen, T., Joutsensalo, J. & Kannisto, L. (2006). Comparison and analysis of the revenue-based adaptive queuing models, *Comput. Netw.* 50(8): 1040–1058.

Sayenko, A., Hämäläinen, T., Joutsensalo, J. & Siltanen, J. (2003). An adaptive approach to wfq with the revenue criterion, *ISCC '03: Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, IEEE Computer Society, Washington, DC, USA, p. 181.

Setton, E., Yoo, T., Zhu, X., Goldsmith, A. & Girod, B. (2005). Cross-layer design of ad hoc networks for real-time video streaming, *IEEE Wireless Commun. Mag.* 12(4): 59–65.

Sheu, S.-T., Lee, Y.-H., Chen, M.-H., Yu, Y.-C. & Huang, Y.-C. (2000). PLFC: The Packet Length Fuzzy Controller to Improve the Performance of WLAN Under the Interference of Microwave Oven, *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE. Volume: 3*, IEEE Communications Society, San Francisco, USA, pp. 1427–1431.

Shreedhar, M. & Varghese, G. (1995). Efficient Fair Queueing Using Deficit Round Robin, *SIGCOMM*, pp. 231–242.

Smadi, M. & Szabados, B. (2006). Error-recovery Service for the IEEE 802.11b Protocol, *IEEE Transactions on Instrumentation and Measurement* 55: 1377–1382.

Tanebaum, A. (1996). *Computer Networks*, 3rd edn, Prentice Hall, New Jersey, USA.

Wang, Z. (2001). *Internet QoS: Architectures and Mechanisms for Quality of Service*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Younis, M., Farrag, O. & D'Amico, W. (2009). Packet Size Pptimization for Increased Throughput in Multi-Level Security Wireless Networks, *Military Communications Conference, 2009. MILCOM 2009.*, IEEE Communications Society, Boston, USA, pp. 1–7.

Zadeh, L. (1965). Fuzzy Sets, *Information and Control* 8: 338 – 353.

Zhang, Q. & Zhang, Y.-Q. (2008). Cross-layer design for qos support in multihop wireless networks, *Proc. of the IEEE, Vol. 96, No. 1*.
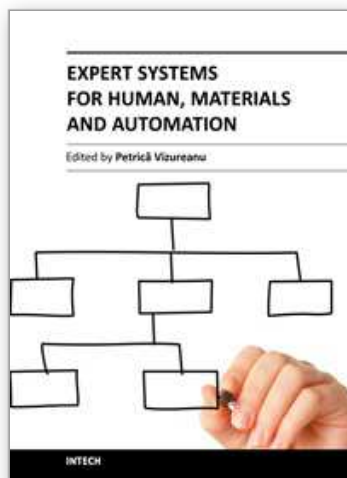
Expert Systems for Human, Materials and Automation

Edited by Prof. Petrică Vizureanu

ISBN 978-953-307-334-7

Hard cover, 392 pages

Publisher InTech

Published online 10, October, 2011

Published in print edition October, 2011

The ability to create intelligent machines has intrigued humans since ancient times, and today with the advent of the computer and 50 years of research into AI programming techniques, the dream of smart machines is becoming a reality. The concept of human-computer interfaces has been undergoing changes over the years. In carrying out the most important tasks is the lack of formalized application methods, mathematical models and advanced computer support. The evolution of biological systems to adapt to their environment has fascinated and challenged scientists to increase their level of understanding of the functional characteristics of such systems. This book has 19 chapters and explain that the expert systems are products of the artificial intelligence, branch of computer science that seeks to develop intelligent programs for human, materials and automation.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tapio Frantti and Mikko Majanen (2011). Fuzzy Based Flow Management of Real-Time Traffic for Quality of Service in WLANs, Expert Systems for Human, Materials and Automation, Prof. Petrică Vizureanu (Ed.), ISBN: 978-953-307-334-7, InTech, Available from: http://www.intechopen.com/books/expert-systems-for-human-materials-and-automation/fuzzy-based-flow-management-of-real-time-traffic-for-quality-of-service-in-wlans

INTECH
open science | open minds