Expert Systems with Applications 41 (2014) 4996–5008

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An expert system for real-time traffic management in wireless local area networks

T. Frantti [a,*], M. Majanen [b]

[a] Renesas Mobile Europe Ltd., Elektroniikkatie 10, FI-90570 Oulu, Finland
[b] VTT Technical Research Centre of Finland, Finland

A R T I C L E I N F O

Keywords:
Expert system
Fuzzy control
Flow control
Offloading
Real-time traffic
WLAN

A B S T R A C T

The paper explores delay-based congestion and flow control and the offloading of real-time traffic from wireless local area networks (WLANs) to mobile cellular networks (MCNs) in multihomed devices. The control system developed is based on an embedded hierarchical expert system. It adjusts transceivers' traffic flow(s) for prevailing network conditions to achieve application-dependent delay and throughput limits. In wireless networks, delay and throughput depend on the packet size, packet transmission interval, and node connection density. Therefore, the controller on the destination node monitors average one-way delay and the change of one-way delay of the incoming traffic. On this basis, it adjusts the packet size and transmission interval of the source node by transmitting a control command to the source. If the prevailing level of traffic in the network exceeds its capacity despite of the control actions taken, devices prepare for developed asynchronous offloading of traffic to another access network.

The control model was validated via simulation of Voice over Internet Protocol (VoIP) traffic in the OMNeT++ network simulator. The results demonstrate that the expert system developed is able to regulate packet sizes to match the prevailing application-dependent optimum and transfer traffic to another network if the network exceed its capacity no matter the control actions taken. Although this work is motivated mainly by issues of congestion and flow control of WLAN systems and the simulations and results were prepared for the IEEE 802.11*b* system, the approach and techniques are not limited to these systems, but they are applicable for other packet switched access networks (PSANs), too.

1. Introduction

With the everyday use of cellular networks, extensive mobility during communication became self-evident reality for end users. A mobility requirement soon appeared also for Internet-based communication, and currently mobile phone networks are moving with Long Term Evolution (LTE) toward an Internet Protocol (IP) infrastructure with the ultimate aim of offering almost all the services provided by current second- and third-generation (2G/3G) cellular networks through an all-IP network (see ITU-T (2004a) and ITU-T (2004b) for more details), although Transmission Control Protocol/Internet Protocol (TCP/IP) networks were not originally designed for mobile use. For example, devices on the Internet are identified by an IP address, which has a dual role: it serves as an identifier and as a locator of the networked device. If a device changes its point of attachment, such as wireless access network, its IP address too may change, which means that other devices in the network need means to access the new address if they are to reach the device at its new location. The change in IP address causes the upper-layer protocol (at the layer above the network layer) connection to break, which is problematic for applications with more persistent connections or applications requiring registration of an IP address.

Today, mobility in IP networks relies on wireless access technologies, and the trend is toward equipping mobile devices, such as smartphones, with multiple network interfaces. The various interfaces are, in the general case, operated by different Internet service providers (ISPs), for devices' improved resilience via an opportunity to connect to the Internet through at least one of the access technologies. For end users, these technologies vary mainly in their coverage area and performance.

A networked mobile device can use the available interfaces either one at a time or several simultaneously. The device using its multiple network interfaces simultaneously is a special case of node multihoming. A multihomed device has multiple IP addresses, assigned to the same network interface or different ones.

* Corresponding author. Tel.: +358 40 547 0819.
E-mail addresses: tfrantti@ee.oulu.fi (T. Frantti), mikko.majanen@vtt.fi (M. Majanen).

http://dx.doi.org/10.1016/j.eswa.2014.01.024
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

Node multihoming differs from site multihoming, which can be defined as an edge network configuration that has more than one service provider but does not provide transit communication between them; see the work of Baker (2011). In site multihoming, end users do not need to manage their devices in any way. When a networked device is multihomed, there are multiple paths between the source and destination devices. The paths may differ in price, data rate, latency, jitter, and packet loss. Node multihoming, with parallel connections, enables new kinds of traffic engineering techniques, whilst site multihoming is intended mainly to increase the reliability of the Internet connection.

In this paper, we explore delay-based congestion and flow control and offloading of real-time traffic from wireless local area networks (WLANs) to cellular networks in multihomed mobile devices. In WLANs, delay and throughput depend on the packet size, packet transmission interval, and connection density. Therefore, we developed and applied control systems to adjust transceivers' packet sizes for prevailing network conditions to achieve application-dependent delay and throughput limits for real-time traffic and to avoid unnecessary offloading. If the prevailing level of traffic in networks exceeds capacity regardless of the control actions, devices prepare to perform asynchronous offloading of traffic to another access network. In our work, the research assumptions can now be presented as: *"Multihomed and multiradio (WLAN and cellular) devices operating in WLAN and running real-time applications try to maximize network capacity by packet size optimization. When the performance threshold is reached, connections are asynchronously offloaded to cellular networks."* By real-time traffic we mean Voice over Internet Protocol (VoIP), video calls, and interactive games. With the assumptions above, the research question for the work can be formulated as follows: *"When should we begin to transfer traffic of real-time applications to another (cellular) network as the WLAN's traffic and number of users increase, and how can we do that to avoid unnecessary offloading?"*

The control model was validated through simulation of VoIP traffic in the OMNeT++ network simulator. Even if this work is addressed mainly to congestion and flow control of WLAN systems, the approach and the techniques are not limited to these systems. They are applicable also for other packet switched access networks (PSANs).

The rest of the paper is organized as follows. Section 2 presents a review of the literature on packet size control in WLANs, multihomed path management, and network selection. Section 3 focuses on multihome- and offloading-related standardization activities. Section 4 summarizes the hierarchical decision making system developed, which features a packet size control unit and real-time offloadig of traffic from WLANs to mobile cellular networks unit. Section 5 describes the simulation model, and Section 6 presents the simulation results. Section 7 delineates our future research. Finally conclusions are drawn in Section 8.

2. Literature review

2.1. Packet size optimization in WLANs

Korhonen and Wang (2005) have studied the effect of packet size on loss rate and delay in an IEEE 802.11-based WLAN. The analysis shows that there is a straightforward connection between packet size, bit error characteristics, and observed delay characteristics. In general, it is evident throughout the literature that the performance of wireless networking is sensitive to packet size and that significant performance improvements are obtained if a "good" packet size is used. For example, Bakshi, Krishna, Vaidya, and Pradhan (1997) show this for TCP traffic over a wireless network. Chee and David (1989), Lettieri and Srivastava (1998), and Chien et al. (1999) all have studied the relationship between frame length and throughput, but they do not propose any precise method for dynamic control of frame length to maximize throughput. Smadi and Szabados (2006) focus on optimization of packet size in error recovery but do not consider the optimal packet size for performance optimization. Sheu, Lee, Chen, Yu, and Huang (2000) present a fuzzy packet length controller (PLFC) for improving the performance of WLANs suffering from interference from a microwave oven. It was demonstrated that the PLFC improves the throughput of User Datagram Protocol (UDP) traffic from that with fixed-length packets, but the authors did not consider performance improvements when the number of users and the amount of traffic are increased. Sankarasubramaniam, Akyildiz, and McLaughlin (2003) have studied packet size optimization for energy efficiency, and Younis, Farrag, and D'Amico (2009) consider it for security and throughput, but their solutions are statistical in nature, meaning that the packet size is optimized beforehand. In the most recent of our publications (Frantti & Majanen, 2011), we presented and compared proportional-integral-derivative (PID) and fuzzy control systems to achieve maximum throughput and minimal delay by adjusting the packet sizes of UDP-based uni- or bi-directional real-time traffic in WLANs in line with prevailing channel conditions. The aim with the hierarchical expert system developed for this paper is to reach application-dependent delay and throughput limits for the maximum number of such real-time connections as VoIP calls and, if necessary, offload real-time connections to another network interface in a controlled manner.

2.2. Multihomed path management

In node multihoming, the end nodes are responsible for paths' management. Fekete (2010) notes that multihoming can be used to enable more advanced traffic engineering or load sharing techniques, such as load balancing and load spreading, by distributing traffic over multiple interfaces and addresses. In load balancing, different traffic flows are sent on different paths. In load spreading, packets belonging to the same flow are sent through different paths. Kandula, Lin, Badirkhanli, and Katab (2008), Singh, Alpcan, Agrawal, and Sharma (2010), and Yao, Kanhere, and Hassan (2009) use quality of service (QoS) parameters for balancing the traffic load over the available network interfaces. In host-centric traffic engineering, this is done by selecting the proper source IP address for outgoing packets and notifying peer nodes about the IP address for incoming packets. In network-centric traffic engineering, this is done within the network through tuning of the routing protocols (see Schiller (2005)).

2.3. Multihomed interface selection

Common approaches to network selection are based on estimated QoS parameters for each available network. Adamopoulou, Demestichas, Koutsorodi, and Theologou (2005), Bari and Leung (2007), Psaras and Mamatas (2011), Song and Jamalipour (2005), Wang, Katz, and Giese (1999), Wilson, Lenaghan, and Malyan (2005), Xing and Venkatasubramanian (2005) and Yahiya and Chaouchi (2009) consider network QoS parameters such as delay and capacity. In the literature, researchers have also proposed more user-centric criteria, among them the power consumption of devices (see Yahiya & Chaouchi (2009) and Petander (2009)) or the cost of network use (see Adamopoulou et al. (2005), Bari & Leung (2007), Wilson et al. (2005) and Alkhawlani & Ayesh (2008)). In some of the aforementioned references, including those of Adamopoulou et al. (2005), Song and Jamalipour (2005), Wang et al. (1999), and Alkhawlani and Ayesh (2008), a user is expected to supply policies and preferences. The approach of Alkhawlani and Ayesh (2008) proposes a combination of fuzzy logic and genetic algorithms for determination of the weight for each criterion. Bari and Leung (2007) use a heuristic classification algorithm to compare the available solutions to the ideal expectations. Song and Jamalipour (2005) have used the analytical hierarchy process to rank the choices. Mehani, Boreli, Maher, and Ernst (2011) consider constraint programming techniques and the use of dedicated solvers for the multihomed flow management problem. They consider application quality metrics rather than network QoS and claim that decisions based on network QoS may not lead to the best user-perceived performance.

2.4. Offloading of traffic

Balasubramanian, Mahajan, and Venkataramani (2010) consider how WiFi access (more details about the WiFi Alliance in http://www.wi-fi.org/) could be used to reduce pressure on the 3G spectrum in a vehicular environment by transferring traffic when this is possible. They noted that the average WiFi (through open access points (APs)) and 3G availability across the cities investigated were 11% and 87%, respectively. The authors also noted that in half of the locations where WiFi is available, its throughput was much less than that of 3G, and WiFi loss rates were higher. The results suggest that straightforward way to combine the two will reduce 3G load by 11% at most, and even that will be to the detriment of application performance. For applications that are extremely sensitive to delay or loss, such as VoIP, the authors developed mechanisms that quickly switch to 3G if WiFi cannot successfully transmit the packet within a certain time. The switching mechanism sends the packet on 3G if the WiFi link layer fails to deliver the packet below a set delay threshold.

Lee and Lee (2012) too consider how to reduce pressure on the 3G spectrum in vehicular environments. They assume that drivers' mobility follows a daily routine and that a given user's pattern of application usage is known. These assumptions are used in decisions to offload data to WLAN systems.

Authors in Qualcomm (2010) note that operators would benefit most from seamless Third Generation/Long Term Evolution (3G/LTE) MCN systems with WiFi offloading by applied to the data traffic that requires no more than best effort and low quality of service. They also note that MCNs should keep some traffic, such as VoIP, on 3G/LTE even when WiFi is available. The work for the paper uses Qualcomm's Connectivity Engine, for a mechanism to provide the device dynamically with the operator's policy for unplanned networks, on-device algorithms for detection of WiFi networks, and a mechanism for seamless vertical handover between MCNs and WiFi.

Dimatteo, Pan, Bo, and Li (2011) propose and evaluate an architecture that migrates data traffic from cellular networks to metropolitan WiFi APs and to a peer-to-peer mobile network. They define the delivery methods for both downstream and upstream, and they highlight suitable application scenarios for the architecture. They assume that a significant quantity of mobile data is delay-tolerant in nature and propose a delay-tolerant networking (DTN) approach for intentional delay to certain bulk traffic before transmission.

3. Multihome related standardization

Multihomed mobile devices are being equipped with multiple network interfaces, each corresponding to a different access technology. With multiple options for access technology comes a management challenge, which is dealt with by various standardization bodies, including the Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), and 3rd Generation Partnership Project (3GPP). In the IEEE (WG, 2009), Media Independent Handover (MIH) provides the necessary access-technology-independent abstractions that are required for optimized vertical handovers in multiple-interface mobile devices. The standardization activities of the IETF and 3GPP are described in Sections 3.1 and 3.2, respectively.

3.1. Multihome standardization in the IETF

Multihoming can be handled at different layers of the protocol stack. For example, multihoming solutions at the network layer are transparent to upper layers, making application development easier. However, the upper layers have greater awareness of the nature of the ongoing traffic. This enables more accuracy in decision-making, *e.g.*, with regard to QoS and also gives space for cross-layer design of multihoming and QoS solutions (see Schiller (2005)).

The Site Multihoming by IPv6 Intermediation Protocol (SHIM6, by Nordmark & Bagnulo (2009)) is an IPv6 multihoming solution that defines an upper layer identifier and actual locator. The Mobility EXTensions of IPv6 (MEXT) working group of the IETF (Tsirtsis, Soliman, Montavont, Giaretta, & Kuladinithi, 2011) develops Mobile IPv6 extension to support multihomed nodes. The base MEXT specification consists of several specifications. Of particular relevance, Wakikawa, Devarapalli, Tsirtis, Ernst, and Nagami (2009) define extensions for Mobile IPv6 (MIPv6) that allow the mobile node to register multiple care-of addresses at the home agent and at the corresponding node. The Locator/ID Separation Protocol (LISP, by Farinacci, Fuller, Lewis, & Mayer (2013)) separates the locator and identifier properties of IP addresses. It supports multihoming by allowing the assignment of multiple routing locators (RLOCs) to the same endpoint identifier (EID) prefix, see Iannone, Lewis, Mayer, and Fuller (2013). Fekete (2010) notes that the Host Identity Protocol (HIP, by Moskowitz, Heer, Jokela, & Henderson (2012)) is not a multihoming protocol in essence, but Nikander, Henderson, Vogt, and Arkko (2008) do define a multihoming extension to it.

At the transport layer, there are multiple protocols that support multihoming capabilities or have them built in. The Stream Control Transmission Protocol (SCTP; Steward (2007)) is a connection- and message-oriented protocol with multihoming capabilities. A connection or an *association* consists of multiple streams. Each association may have several source and destination addresses, and only one (source address, destination address) pair is used at a time. If a failure occurs, a different address pair is selected. Steward (2007) and Steward, Tuexen, Poon, Lei, and Yasevich (2011) define primitives for an upper-layer protocol that enable specifying the source-address set and the destination address for transmitted packets, changing the primary path, and retrieving the current primary path and other association information. Steward, Xie, Tuexen, Maruyama, and Kozuka (2007) define an extension that lets end nodes dynamically add addresses to an association or remove them. The extension enables SCTP to react to changes in the address sets. It also allows the end node to note its peer from the preferred receiving address.

MultiPath TCP (MPTCP) supports load spreading. A multihomed device can use MPTCP by setting up a multiple-subflow connection with its peer. Each subflow has a different IP address pair (Ford, Raiciu, Handley, & Bonaventure, 2012). Scharf and Ford (2012) propose application-interface extensions for MPTCP that let applications enable/disable MPTCP, define the set of addresses and network interfaces for subflows, and obtain information about the current subflows. The Multihoming for Datagram Congestion Control Protocol (DCCP Kohler, Handley, & Floyd, 2006) is provided via a combination of separate connections to the main connection, see Kohler, Handley, and Floyd (2006).

3.2. Multihome standardization in the 3GPP

Selection of network to choose the optimal access technology cannot always be fully decoupled from the selection of the service provider. The 3GPP specifies the rules for selecting a service provider (operator) very strictly in GPP TS 23.122 (2011). The challenge for a WLAN-capable 3GPP device is to use the most appropriate access technology without violating those rules for service-provider selection that are related to its cellular subscription.

Legacy and current cellular standards provide IP addressing of devices and interoperable tools to resolve the issues of mobility, security, QoS, and charging in each operator's network. Emerging LTE provides an all-IP core architecture, which allows IPv6 and dual stack IPv4/IPv6 connectivity and provides mobility between 3GPP networks and non-3GPP and legacy networks. The application level of multihoming is not within the scope of 3GPP, but several enablers for multihoming are under study as ongoing work items and addressed in recently completed 3GPP specifications. These include the means for a network operator to optimize the non-3GPP access discovery by the terminal via Access Network Discovery and Selection Function (ANDSF) in GPP TS 24.312 (2011) and 3GPP-WLAN interworking via GPP TS 24.235 (2011). Multi-Access Packet Data Network Connectivity (MAPCON) addresses multihoming paradigms for multiple paths. Support for multihoming and simultaneous multiple access in LTE is also likely to be handled through the efforts of the IETF MEXT working group. Enablers for offloading selected traffic flows to non-3GPP access have been specified by several 3GPP groups under the work items "IP Flow Mobility and Seamless WLAN Offloading" (IFOM) and "Selected IP Traffic Offload" (SIPTO) as part of 3GPP Release 10.

4. A decision making system for real-time traffic management

The decision making system developed is based on a hierarchical two-layer embedded expert system; see Fig. 1. The higher layer (*Decision making logic for real-time traffic management*) is used to trigger actions in the lower layer modules (*Packet size control* and *Traffic off-loading*), *i.e.*, it decides whether to prepare to offload traffic or to adjust transceivers' traffic flow(s) for prevailing network conditions by means of the controllers developed.

Initially, the decision-making logic directs control to the packet size control unit and allows it retain control of its flow(s) at least for a settling interval. The average settling time for the fuzzy packet size controller was about 32.50 s; see Table 4. If the delay time or throughput after the settling time remains above the threshold value(s) for more than three times the worst-case transient response time of 14.48 s (see Section 6.3), the decision-making logic triggers traffic offloading functions. The decision making logic for real-time traffic management has a rule *IF delay IS more than the threshold value OR throughput IS NOT that required THEN proceed to traffic offloading*. The traffic-offloading unit monitors control signals in its coverage area to determine whether any other node has already initiated an offloading (external offloading) or disconnection procedure, before it itself proceeds to do offloading (internal offloading). It operates according to the rules *IF an external offloading or disconnection process IS ongoing, THEN defer internal offloading AND return control to the decision-making logic unit* and *IF no external offloading or disconnection is observed, THEN proceed to internal offloading AND return control to the decision-making unit*.

The PID and fuzzy packet size controllers developed are used at the destination nodes to change real-time traffic packet payload size on the application level. In our system, all destination nodes monitor congestion by measuring *average one-way delay error*

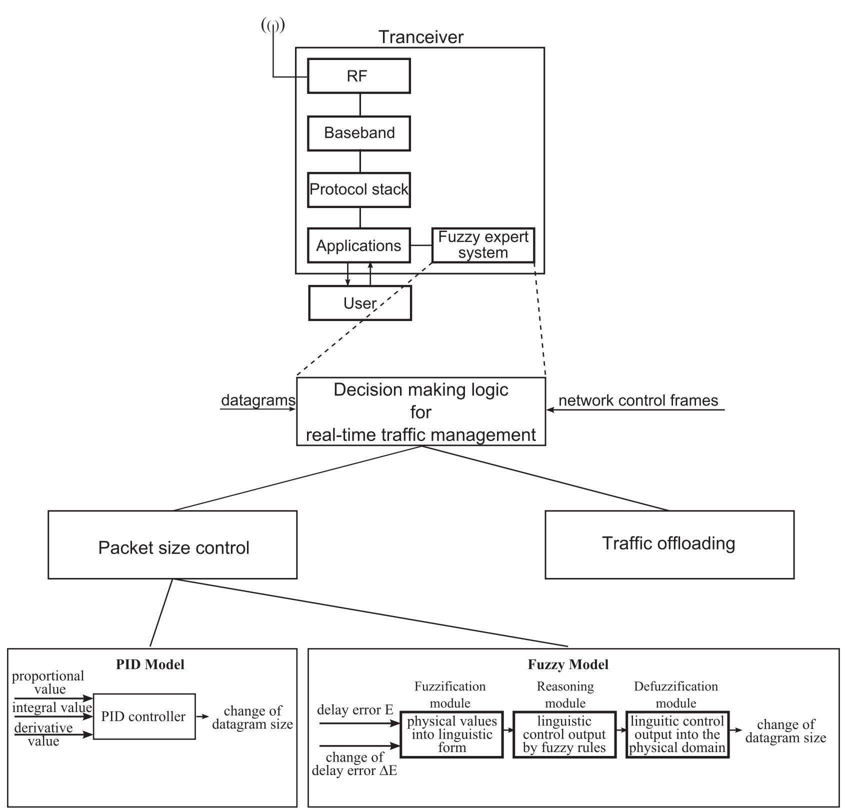

Fig. 1. Decision making logic for real-time traffic management.

and the *change in one-way delay error*, set packet size incrementing on their basis, and deliver packet size information to the source node. Both controllers were designed to reach an application-dependent target end-to-end delay and exact throughput level in the prevailing channel conditions.

The traffic offloader is used at the destination nodes to handle offloading of traffic to another access network (an MCN). The offloader is activated if the prevailing level of traffic in the WLAN exceeds its capacity regardless of the control actions. It monitors network control frames to see whether any of its neighboring nodes indicates preparation to offload traffic. If the offloader detects that a neighboring node is preparing for offloading, the offloader defers its own offloading process to determine the effects of the prevailing neighbor node's offloading on the delay time. If the delay time does not drop enough, the node proceeds to the offloading process. Otherwise, the decision-making logic transfers control to the packet size controller.

4.1. Proportional-integral-derivative controller

A proportional-integral-derivative (PID) control is a widely used feedback control mechanism. It is used here as a comparative control method. A PID controller calculates an error value, 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 from the sum of recent errors, and the *derivative* value determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust elements of the process, such as the transmitter's packet payload size. In the PID controller developed, one-way delay error (E_d = proportional value = delay − target value), sum of recent errors (I_d = integral value), and change in error (ΔE_d = derivative value) are used as the input values. The output value of the controller is the change in packet payload size. The new packet payload size is the change in the packet payload size + the earlier packet size. In Frantti and Majanen (2011) we have successfully used the controller to reach a set application-dependent target end-to-end delay with the maximum throughput in the prevailing channel conditions. Maximum throughput instead of the fixed minimum required throughput with strict end-to-end delay is needed, for example, in video conversations with scalable video coding.

This controller can be presented in 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 in 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 literature contains proposals of diverse types of tuning rules, among them the Ziegler–Nichols educated guesses at the parameter values. Most of these tuning rules provide a starting point for the on-site fine-tuning of PID controllers. In addition, automatic tuning methods have been developed for PID controllers, and this is one of our research targets for the future too. Here the parameter values were defined through application of both main Ziegler–Nichols approaches as a starting point for the experimental fine-tuning. The Ziegler-Nichols approaches are introduced by, for example, Ogata (2009) in detail.

4.2. Fuzzy controller

Fuzzy control was originally developed to include input of a human operator or system engineer's expertise, which does not lend itself to being easily expressed in differential equations so much as in situation/action rules. In the fuzzy control system developed here, 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* (E_d) and the *change in one-way delay error* (ΔE_d), and the output value is the *packet size increment*. This is referred to as a *two-input, single-output control strategy*. The structure of the controller located at the user terminal is presented in Fig. 1.

The fuzzyfication procedure is illustrated in Figs. 2 and 3. In Fig. 2, delay error E_d is −24.92 ms, which is *negative big* at grade 0.48 and *negative small* at grade 0.52. The change in delay, ΔE_d, is 6.46 ms, which is *zero* at grade 0.77 and *positive small* at grade 0.23, see Fig. 3.

A rule base includes a control policy, which is usually presented with linguistic conditional statements, *i.e.*, if-then rules. In this application, a linguistic model of a system was described by linguistic relations. The linguistic relations form a rule base (25 rules, see Fig. 5) that was converted into numerical equations to decrease the computation load of the controller. Suppose, as an example, that X_{ij}, $i = 1,2;\ j = 1,\ldots, m$ (j is an uneven number) is a linguistic level (*e.g.,negative big, negative small, zero, positive small, and positive big*) for the variable X_i. The linguistic levels are replaced with integers $\frac{-(j-1)}{2}, \ldots, -2, -1, 0, 1, 2, \ldots, \frac{(j-1)}{2}$. The direction of the interaction between fuzzy sets is presented by coefficients $A_{ij} = \{-1, 0, 1\}$, $i = 1,2;\ j = 1,\ldots, m$. 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, see Juuso (1993). Thus a compact equation for output Z_{ij} is

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

The mapping of linguistic relations to linguistic equations for this application is described in Fig. 5. For example, we can read from

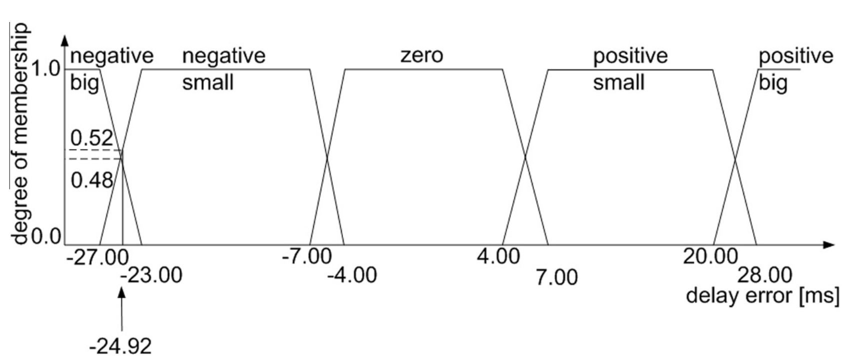

Fig. 2. Fuzzy membership functions for the E_d.

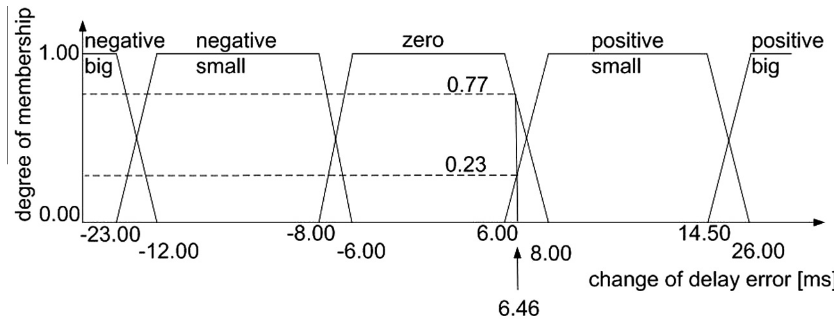

Fig. 3. Fuzzy membership functions for the ΔE_d.

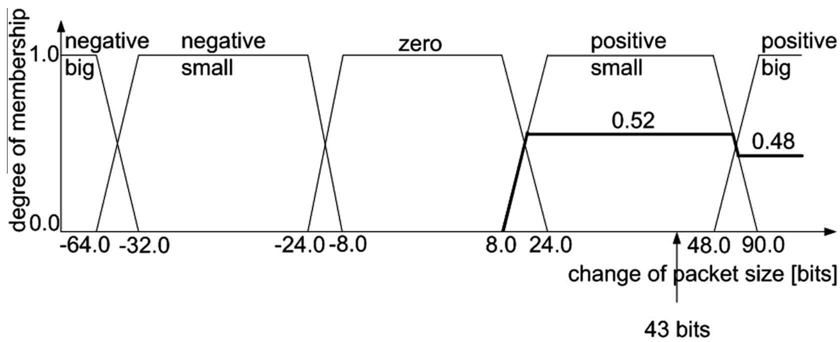

Fig. 4. Fuzzy membership functions for the change of packet size.

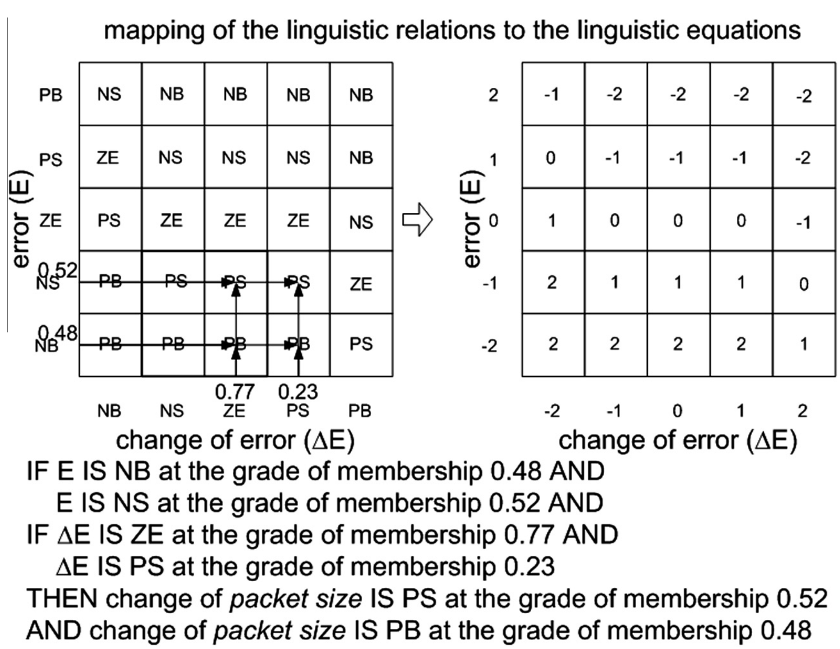

Fig. 5. Fuzzy rule base and mapping of the linguistic relations to the linguistic equations.

Fig. 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 $\lceil\frac{(-1*-1+-1*0)}{2}\rceil = 1$. More detailed reasoning examples are provided by, for example, Frantti and Majanen (2011) and Frantti (2012). On the other hand, the higher-layer decision-making logic for real-time traffic management (see the beginning of Section 4 and Fig. 1, above) that is used to trigger actions in the lower-layer modules, is presented by if-then rules.

The reasoning process produces the linguistic control output, *packet size increment*, which is transformed back into the physical domain in the defuzzification phase for finding of the crisp control output value for the change in packet size. In the defuzzyfication phase, the center-of-area (CoA) method was used. The defuzzyfication procedure is illustrated in Fig. 4, from which it can be seen that the change in packet size is *positive small* at grade 0.52 and *positive big* at grade 0.48. The crisp output value represents the center of the area, *i.e.*, the new packet size is 43 bits above the earlier value. The control command is transmitted to the source node by acknowledgement packets.

4.2.1. Adaptation

To adapt to application-dependent delay and throughput requirements, the expert system developed needs as inputs the target application-dependent delay and throughput values. For VoIP traffic, the target delay is set to the maximum acceptable value, and target throughput is an exact value. Control of real-time traffic flow(s) then proceeds through optimization of packet sizes and thereby packet transmission intervals for the target delay and throughput in the prevailing channel conditions. If the level of network traffic exceeds capacity no matter the devices' control actions and the delay increases or throughout decreases for some node(s), the higher-layer decision-making logic of the relevant node triggers traffic offloading functionality and the device prepares to offload traffic from WLAN to MCN.

To keep the membership functions and inference logic independent from absolute dependency on delay value and packet size, the expert system uses relative input values (*delay error* and the *change in delay error*). The expert system also determines the increment of the packet size as an output value, instead of the absolute packet size. With an absolute input variable, such as *delay*, the membership functions should be redefined for all possible target delay values.

4.2.2. Computational complexity

The expert system for real-time traffic management at the mobile nodes increases the computational complexity for the node in question. The implementation decision for the system involves a tradeoff among complexity, required computation time, the random access memory (RAM) and program memory needed, and the advantages gained by means of the algorithm. It increases the computational load when deciding whether to prepare to offload traffic or to adjust transceivers' traffic flow(s) for prevailing network conditions via the controllers developed. On the other hand, it decreases computational load by preventing futile control actions and unnecessary offloadings.

The decision-making logic performs two comparisons after receipt of every 21 packets, one for the delay threshold and one for the throughput value, to apply the rule *IF the delay IS more than the threshold value OR throughput IS NOT that required, THEN proceed to traffic offloading*. The traffic offloading unit executes a comparison to decide whether to defer or proceed to the offloading process via the rules *IF external offloading or the disconnection process IS ongoing, THEN defer internal offloading AND return control to the decision-making logic unit* and *IF no external offloading or disconnection is observed, THEN proceed to internal offloading AND return control to the decision-making unit*.

The computational load of the scanning and of the disassociation and disconnection processes are not taken into account, because they are IEEE 802.11 standardized solutions, see SubSection 4.3. Each network node within radio-coverage range listens to the transmission and uses the first address in the 802.11 WLAN MAC header to determine whether to process it. The MAC header states also the type of the frame, enabling the nodes to see the initiation of disassociation or disconnection procedures by neighboring nodes. Recording possible disassociation or disconnection information for further use requires one comparison and one write and one read operation. The comparison and the write operations are performed when disassociation or disconnection is observed. From the energy consumption point of view, these occur so infrequently that they can be ignored. The read operation is needed after receipt of every 21 packets. When the read operation occurs, the memory content read must be erased, too.

The fuzzy feedback control also slightly increases communication load, by transmitting application-level acknowledgements after every 21 received packets. The fuzzyfication phase requires, at the most, 2×9 comparisons, 2×2 adds, and 2×1 multiplies. In the comparisons, crisp input values are compared to the parts of the membership functions that cover the dynamic ranges of the input variables, see Fig. 2. After the comparisons, in which identification of the fixed fuzzy label(s) occurs, the degree of membership is determined via multiplication of the interval between the corner point and crisp value by the angle of the line. Multiplication is not needed if the crisp point is in the upper area, *i.e.*, if the degree of membership is 1.0. The reasoning process for linguistic equations requires no more than eight multiplies, four additions, four divisions, and six min./max. comparisons. The defuzzyfication phase requires at most 2×2 additions and 2×5 multiplies for definition of the horizontal component of the center of area. All in all, the control method developed adds no more than 56 computations for the definition of change in fuzzy packet size.

According to Koomey (2010) it can be estimated that one operation requires 1.2–1.8 nJ, which means that 56 operations of the fuzzy feedback control requires about 56×1.5 nJ $= 84.0$ nJ if the value of 1.5 nJ/operation is used. The estimate of the energy consumed by the fuzzy controller per transmitted packet is then $\frac{84.0}{21}$ nJ $= 4.0$ nJ, which is less than the energy consumption of three elementary operations. In addition, the decision-making logic and offloading unit require, at most, four operations for a transmitted packet, using $\frac{6.0}{21}$ nJ = 0.29 nJ.

4.3. Traffic off-loading methods

An association request frame of a WLAN station enables an access point to allocate resources for it, synchronize with its radio network interface card (RNIC), and establish an association ID for the RNIC. The station can send a disassociation frame to the access point if it wishes to terminate the association. The access point can then release the memory allocations and remove the RNIC from the association table.

An IEEE-802.11x-based WLAN node continually scans all the 802.11 allocated radio channels to see whether the channel is free or if the frame received is aimed for it. If a node transmits and the channel is not available, it defers the transmission. However, it does not have a mechanism to transfer (offload) real-time traffic to another type of network in a controlled manner. Here, the traffic offloader developed is used at the destination nodes to offload traffic to another access network (MCN).

Delay times of WLAN nodes increase and approach the application-specific delay time thresholds set as the number of users and the amount of traffic in the AP's coverage area increase. This may lead to congestion and parallel disconnections. In our model, the offloader is activated if the prevailing level of traffic in the WLAN network exceeds the network's capacity no matter the control actions and the nodes perform the offloading or disconnection sequentially to maximize the efficient use of WLANs.

In the model, the offloading or disconnecting node informs its neighbors that it wants to terminate a connection. This was done by means of disassociation frames. For example, if the node wants to disassociate, it can transmit a disassociation frame to its AP. Nodes nearby, themselves preparing to disassociate because of the increased delay time, could then defer the disassociation process to gauge the effects of the present disassociation on the delay time. If the delay time does not drop enough, the next node proceeds to the disassociation process; however, if the delay time falls enough, the decision-making logic transfers control to the packet size controller and the nodes continue prevailing real-time connections. If the node wants to disconnect but preserve its association, it cannot use a disassociation frame as such. Instead it transmits the disassociation frame without the source node address and uses the broadcast address as a destination address. Nodes nearby then defer their offloading or disconnection in order to assess the effects of the neighboring node's disconnection on the delay times.

5. Network simulations

The simulations were performed with the OMNeT++ 4.0 simulator (see http://www.omnetpp.org) with the INETMANET framework. The simulation model consisted of up to 48 wireless hosts in infrastructure (BSS) mode and one IEEE 802.11*b* WLAN access point. The nodes were distributed randomly around the AP, and the distance from a host to the AP varied between 14.1 and 77.8 m. The nodes were not moving, and it was assumed that they were synchronized by means of, for example, the access point's beacon message for determination of the exact delay time.

In the simulation scenario, the performance of VoIP traffic was studied. The VoIP traffic's data rate was 64 kbit/s, corresponding to an uncoded pulse code modulated (PCM) voice signal. The fixed packet size used was 160 octets (1280 bits), *i.e.*, the packet interval was 20 ms. With the controllers, the initial packet size was 50 octets, for a 6.25 ms packet interval. The VoIP calls were made in pairs, with host0 and host1 forming a pair, host2 and host3 another pair, etc. All hosts measured the delay for the packets, used our packet size optimization algorithms to calculate the optimal packet size for 150 ms (or even 200 ms, see SubSection 6.1) target delay, and reported it back to each peer after every 21 packets via a control command message (piggybacked on the acknowledgement packet) over UDP. Then the pair adjusted its packet size. Also, the packet interval was adjusted, to keep the data rate constant. The control message interval of 21 packets (the time interval was $21 \times [6.25\text{ ms} - 150\text{ms}] = [131.25\text{ ms} - 3150\text{ ms}]$ depending on the used packet size) was experimentally evaluated (see SubSection 6.1) from the rise- and settling-time responses and its optimization remains for our future research.

6. Results

The decision making system developed, with the flow controllers, was designed for interactive real-time applications such as VoIP calls, video calls, and interactive games, to reach the application-dependent target end-to-end delay and throughput values and also maximize the number of real-time connections in an access point's coverage area. The simulation scenarios employed measured delay and throughput of VoIP traffic when the packet payload size was fixed, adjusted by the PID controller developed, and managed by the fuzzy controller we designed.

The delay and throughput evaluations were performed as a function of changes in the number of VoIP connections. Our results in an earlier publication Frantti and Majanen (2010) showed that there is an optimal packet size with respect to the overall delay and packet loss rate, one that depends on the number and type of real-time connections. However, the amount of background traffic changes as a function of time and it is not possible to choose an optimal fixed packet size manually in response to the fluctuating level of background traffic, so the options are either to keep it constantly the same level, *e.g.* on 160 octets (1280 bits), or to use automatic packet size controllers.

6.1. Throughput and delay

The application-level one-way throughput for uncoded, non-compressed G7.11 PCM VoIP conversation requires 64 kbit/s. Uncoded, non-compressed G7.11 PCM is used for minimal delay. However, depending on the voice-sample sizes, this may demands even a 107.20 kbit/s link layer data rate with 10 ms voice samples (1000 ms/10 ms $\times$ [80 octets for voice samples + 54 octets for packet header = 134 octets = 1072 bits link layer packet size] = 107.20 kbit/s). For 50 ms voice-sample sizes this rises to 72.64 kbit/s (1000 ms/50 ms $\times$ [400 octets + 54 octets = 454 octets = 3632 bits packet size] = 72.64 kbit/s).

The overall delay in VoIP calls includes delays at the MAC layer, at the link layer, and at the TCP/IP protocol stack; propagation delay in the radio channel; queueing delays in intermediate nodes; speech-coding delays; delay for the jitter (packet-delay variation) buffer; and the look-ahead delay of the codec. Here, the size of the jitter buffer was varied from 20 ms to 100 ms with the packet payload sizes. This buffering delay is introduced before the playout of the mediastream begins, *i.e.*, the receiver delays the time at which it starts playing back the mediastream. For larger packets, the jitter-buffer time was shorter than for smaller ones, on account of the longer delay caused by speech coding for larger packets in order to keep the overall delay for all the packets below 150 ms. According to Andrews, Ghosh, and Muhamed (2007) and ITU-T G.114 (ITU-T, 1993), absolute delay should not exceed 150 ms for good quality of voice communication.

To avoid unnecessary offloading, we relaxed the absolute delay requirement and soft offload threshold for VoIP conversations to 200 ms (offloading was to be done when the delay is *around* the 200 ms). The quality of voice deteriorates as delay increses, but the communication is still understandable when there is very low packet loss rate. Using a 200 ms soft offload threshold allows a rouhly 50 ms hysteresis range for offloading and load balancing among networks. Therefore, the delay from the jitter buffer and speech coding together was designed to be a constant 175 ms, consisting of 20–100 ms delay from the jitter buffer, 20–150 ms delay for audio samples' coding, and the 5 ms look-ahead delay of the speech-coding algorithm. Because of the constant 175 ms delay, the protocol, queuing, and propagation delay should remain below 25 ms in total if the 200 ms delay limit is to be met. This allows even a 1254 octet link layer packet size, because the application data rate = 64 kbit/s = 8000 octets/s = 8 octets/ms and a 150 ms sample size means 150 ms $\times$ 8 octets/ms = 1200 octet application-level packet size. When we add the 54 octets of the packet header, we obtain an overall packet size of 1254 octets. Fig. 6 presents protocol, queuing, and propagation delay as a function of the number of nodes for a fixed packet size of 1280 bits, for the fuzzily controlled flows, and for the PID-controlled flows. The delay increases rapidly from 12 nodes onward when this fixed packet size is used. With the PID and fuzzy controllers, the delay increases slightly until more rapid increases occur after 38 and 42 nodes are reached, respectively. The increase in delay as a function of the number of nodes, mainly due to increased media access (queuing) delay, and the rapid increase in delay predict imminent congestion.

Tables 1–3 present average delay (protocol + queuing + propagation delay), throughput, number of connections for which the delay limit was exceeded, and number of connections that did not reach the throughput threshold, for the fixed packet size of 1280 bits, for the fuzzily controlled flows, and for the PID-controlled flows, with different numbers of VoIP connectionpairs. With the fixed packet size of 1280 bits, delay and throughput requirements were met for up to 12 nodes. The limiting factor with the 1280 bit packet size was throughput. Hence, it would be possible to increase the number of nodes slightly by reducing throughput and increasing delay with voice-sample compression. Fig. 7 presents the development of throughput as a function of the number of nodes for the fixed 1280 bit packet size, for the fuzzily, and PID-controlled flows. Connections were initiated to the network one by one during the first 300 s, to avoid congestion due to address-resolution handshakes in the beginning of several parallel or nearly parallel connection initiations.

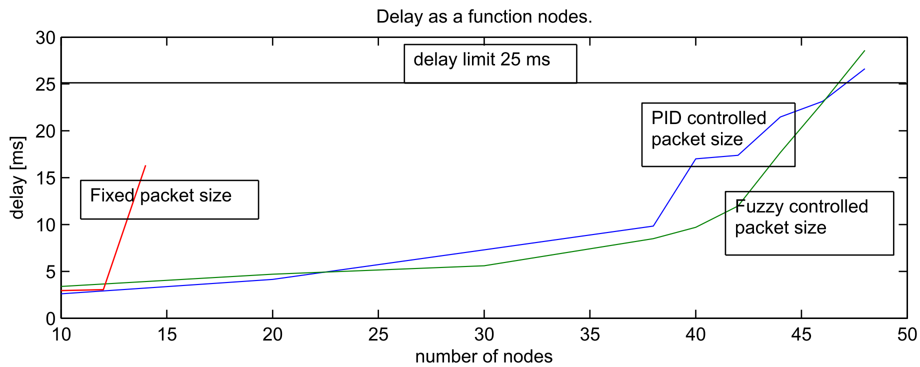

Fig. 6. Queuing and propagation delay as a function of number of nodes.

Table 1
Average delay, throughputs, number of connections past the delay threshold, and number of connections that did not reach the throughput threshold, when a fixed packet size of 1280 bits was used. The protocol, queuing and propagation delay limit was 25 ms, and the throughput limit 64 kbit/s. VoIP traffic.

Number of nodes	Delay [ms]	Throughput [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]	Long settling time[No. connections]
12 Nodes	3.05	64.00	0	0	0
14 Nodes	16.32	54.32	14	0	0

Table 2
Average delay, throughputs, number of connections past the delay threshold, and number of connections that did not reach the throughput threshold, when the fuzzy controller was used. The protocol, queuing and propagation delay limit was 25 ms, and the throughput limit 64 kbit/s. VoIP traffic.

Number of nodes	Delay [ms]	Throughput [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]	Long settling time [No. connections]
36 Nodes	7.90	64.00	0	0	0
38 Nodes	8.50	64.00	0	0	0
40 Nodes	9.70	64.00	0	0	0
42 Nodes	11.97	64.00	0	0	6
44 Nodes	17.68	64.00	0	5	8
46 Nodes	22.99	63.09	17	10	9
48 Nodes	28.59	58.78	0	n/a	n/a

Table 3
Average delay, throughputs, number of connections past the delay threshold, and number of connections that did not reach the throughput threshold, when the PID controller was used. The protocol, queuing and propagation delay limit was 25 ms, and the throughput limit 64 kbit/s. VoIP traffic.

Number of nodes	Delay [ms]	Throughput [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]	Long settling time [No. connections]
36 Nodes	7.30	64.00	0	0	0
38 Nodes	9.84	64.00	0	0	2
40 Nodes	17.01	63.89	5	6	4
42 Nodes	17.39	61.58	20	12	9
44 Nodes	21.48	60.65	n/a	13	16
46 Nodes	23.14	58.93	n/a	15	n/a
48 Nodes	26.63	55.861	n/a	20	n/a

The average delay was acceptable for up to 46 nodes, and the requirement for average throughput was reached for up to 44 when the fuzzy controller was used, see Figs. 6 and 7. However, the limits for delay and settling times were exceeded slightly with five and eight nodes, respectively, as shown in Table 2. For up to 42 nodes, the delay and throughput limits were achieved perfectly at all nodes, although there was minor exceeding of the threshold for settling time.

When the PID controller was used, the average delay was acceptable with up to 46 nodes but throughput limits were met up to 38 nodes (see Figs. 6 and 7) although the settling time was overly long for two connections (see Table 3). With up to 36 nodes, all requirements were met.

It can be observed from Figs. 6 and 7 and Tables 1 and 2 that the controllers developed clearly increase the number of connections in comparison to the typical 1280 bit fixed packet size. The fuzzy controller enables about six nodes more than the PID controller.

6.2. Response times

The simulations conducted measured also the average rise and settling times (averaged over the connections) of the controllers, see Tables 4 and 5. The average rise and settling times for the fuzzy controller with the maximum number of connections (44 nodes) were 18.82 s and 32.50 s, respectively. For the PID controller with 38 nodes, the average rise time was 9.29 s and the settling time

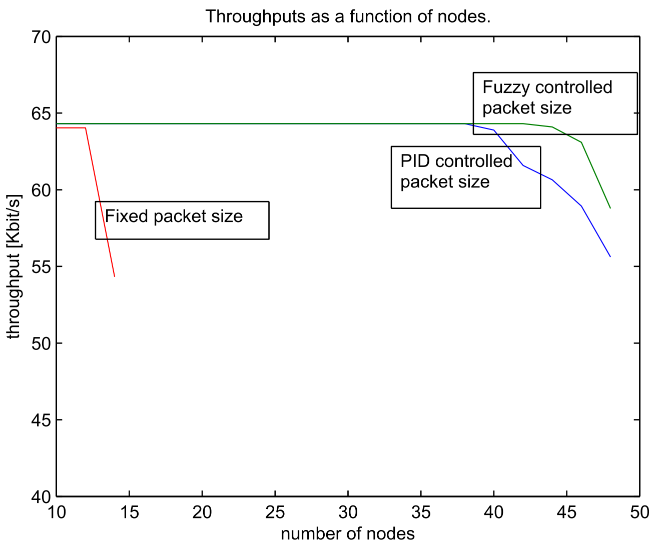

Fig. 7. Throughput as a function of the number of nodes.

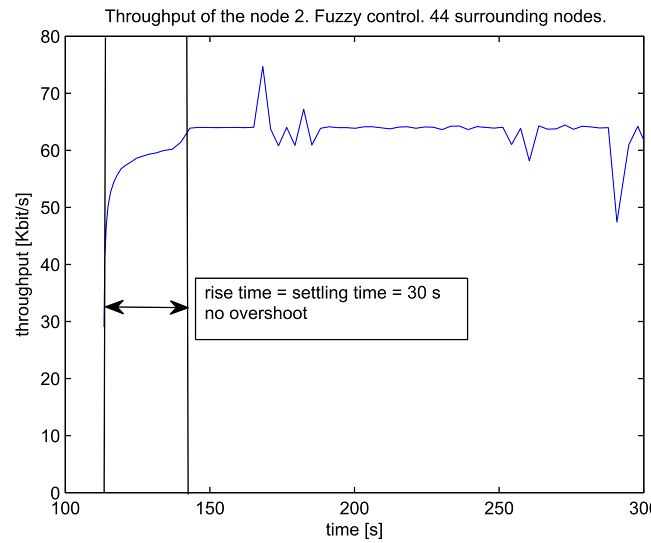

Fig. 8. Rise and settling times of the node 2 when fuzzy controller is used. 44 Nodes.

Table 4
Average rise and settling times of fuzzy controller.

Number of Nodes	Rise time [s]	Settling time[s]
36 Nodes	17.38	17.38
38 Nodes	17.56	23.50
40 Nodes	18.17	24.50
42 Nodes	18.43	24.29
44 Nodes	18.82	32.50
46 Nodes	20.82	42.29
48 Nodes	21.21	54.89

Table 5
Average rise and settling times of PID controller.

Number of Nodes	Rise time [s]	Settling time [s]
36 Nodes	9.19	20.18
38 Nodes	9.29	23.42
40 Nodes	11.62	38.50
42 Nodes	17.39	44.29
44 Nodes	18.82	55.59
46 Nodes	19.60	77.13
48 Nodes	22.51	82.63

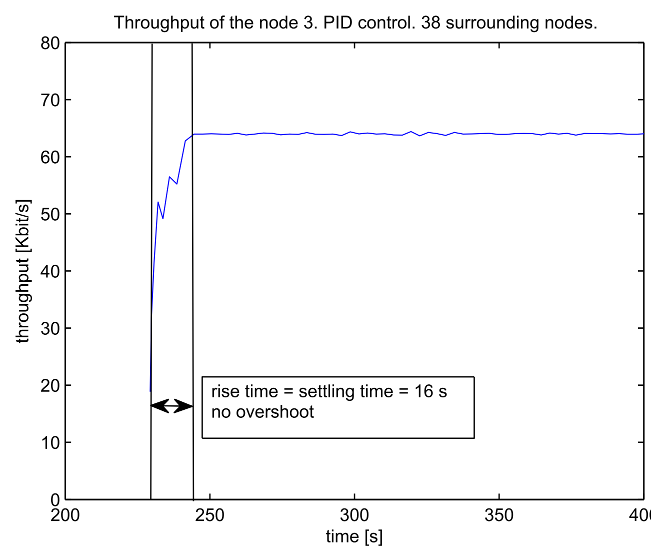

Fig. 9. Rise and settling time of the node 3 when PID controller is used. 38 Nodes.

23.42 s. Figs. 8 and 9 illustrate the rise and settling times of example nodes when the fuzzy and PID controllers are used. The peaks in the throughput curves are probably due to temporal congestion situations, wherein some packets are lost and the prevailing irregular transmission-time interval in congestion is compensated for through either packet size adjustment or packet bursts.

6.3. Effects of off-loading

Tables 6–8 present average delay change (protocol + queuing + propagation delays), throughput change, number of connections with delay value exceeded, and number of connections that did not reach the throughput threshold when one connection (two nodes) was offloaded for the fixed packet size 1280 bits, for the fuzzily controlled flows, and for the PID-controlled flows. Average delay decreased and throughput increased 9.68 kbit/s to the required 64.00 kbit/s when the fixed packet size was used, see Table 6. With fuzzily controlled flows, the average delay also decreased and throughput increased 0.91 kbit/s to the required 64.00 kbit/s, with all the nodes reaching the target values set for delay and throughput. From Table 2 we can see that there was slight exceeding of the delay values for five connections when there were 44 nodes. However, the exessive delays were eliminated after decreasing of the number of nodes from 46 to 44, see Table 7. This is probably because there is a different set of nodes remaining after the offloading. In Table 2, the nodes 1–44 participate whereas in Table 7 nodes 1, 2, and 5–46 remain after the offloading, *i.e.*, nodes 3 and 4 have been offloaded. With PID-controlled flows, the average delay decreased and throughput increased by 0.11 kbit/s to the required 64.00 kbit/s and all nodes reached the target values for delay and throughput, see Table 8.

Figs. 10 and 11 illustrate transient response times of the fuzzy and PID controllers when nodes 2 and 3 leave after 400 s of simulation time and nodes 6 and 7 at 500 s. After the first offloading, at 400 s, the transient response times were about 5 s for both controllers. After the second offloading transients cannot be observed. This may be because the system already has enough bandwidth for each connection.

Table 6
Effects of the two-node offloading (from 14 to 12 nodes, with nodes 3 and 4 offloaded) on the observed VoIP conversation with a fixed packet size of 1280 bits.

Offloading	Delay change [ms]	Throughput change [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]
From 14 to 12 nodes	−13.27	9.68	−14	0

Table 7
Effects of the two-nodes offloading (from 46 to 44 nodes, with nodes 3 and 4 offloaded) on the observed VoIP conversation with fuzzily controlled packet sizes.

Offloading	Delay change [ms]	Throughput change [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]
From 46 to 44 nodes	−10.32	0.91	−17	−10

Table 8
Effects of the two-nodes offloading (from 40 to 38 nodes, with nodes 3 and 4 offloaded) on the observed VoIP conversation with PID-controlled packet sizes.

Offloading	Delay change [ms]	Throughput change [kbit/s]	Low throughput [No. connections]	Delay value exceeds [No. connections]
From 40 to 38 Nodes	−3.62	0.11	−5	−6

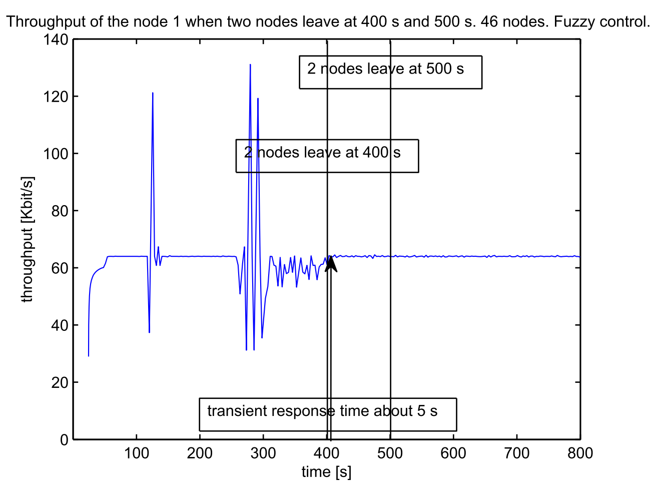

Fig. 10. Throughput of the node 1 when fuzzy controller is used and two nodes leave at 400 s and at 500 s. 46 Nodes.

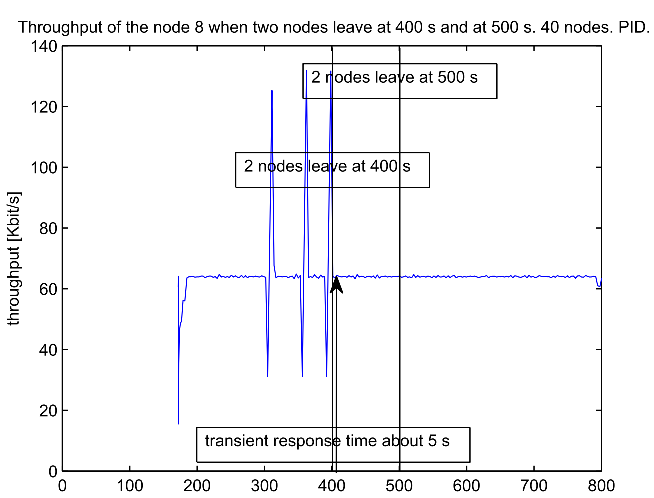

Fig. 11. Throughput of the node 8 when PID controller is used and two nodes leave at 400 s and at 500 s. 40 Nodes.

The average transient response times of the fuzzy and PID controllers after the offloading from 46 to 44 nodes and 40 to 38 nodes, were about 4.8 s. The transient response time is much shorter than the settling time (32.50 s as compared to 23.42 s) because the packet size is already tuned to near the optimal value before the transient situation. Settling and transient times can also be decreased via an increase in the control-command frequency, which in our case was one command after receipt of 21 packets.

To improve the transient responses, we also estimated a random time interval $j \in [2 \times average\ transient\ response\ time = 9.65\ s, 3 \times average\ transient\ response\ time = 14.48\ s]$ that nearby stations should wait after observing external offloading before they begin internal off-loading due to increased delay time. From the application standpoint, the waiting time could be even longer, because quality thresholds for such applications as VoIP conversations are not absolute: the quality deteriorates gradually as a function of delay time.

7. Future research

Our research plans include deeper integration of the traffic-offloading logic and distributed packet size control logic described here. The aim is to develop a decision-making unit that autonomously adapts to the minimum random time interval that nearby stations should wait before they reconsider traffic offloading if, for example, the packet size control frequency changes.

8. Conclusions

In this paper we explored delay-based congestion and flow control along with offloading of VoIP traffic from WLANs to MCNs. The decision making system developed with the controllers applied is based on an embedded hierarchical expert system. It regulates source nodes' packet sizes for prevailing network conditions in order to reach the target throughput and pass beneath the maximum allowed delay for VoIP traffic in WLANs, as much as to avoid unnecessary offloading. If the prevailing level of traffic in a network exceeds capacity regardless of the control actions, devices prepare to perform asynchronous offloading of traffic to another access network.

The decision making systems are at user terminals. The models were validated through simulation of VoIP traffic over UDP in a WLAN environment with an OMNeT++ network simulator. The results showed that the decision making system developed was able to set packet payload size values to the prevailing optimal level very quickly and accurately, and they also managed asynchronous offloading of traffic to another access network when the relevant network's traffic exceeded its capacity. The models developed enable WLANs to increase the number of concurrent users to three or four times those seen with fixed packet sizes and to improve the quality of the connections with minimal delay. The random time interval that nearby stations should wait before they reconsider traffic offloading was evaluated to be about 9.65 s–14.48 s.

Acknowledgment

The research was supported by Renesas Mobile Europe Ltd.; VTT Technical Research Centre of Finland; and the Internet of Things program of DIGILE, funded by Tekes.

References

3GPP TS 23.122. (2011). 3rd Generation parnership project; technical specification group core network and terminals; non-access-stratum (NAS) functions realted to Mobile Station (MS) in idle mode (Release 10). Tech. Rep., 3GPP.

3GPP TS 24.235. (2011). 3rd Generation parnership project; technical specification group core network and terminals; 3GPP system to wireless local area network (WLAN) interworking management object (MO) (Release 10). Tech. Rep., 3GPP.

3GPP TS 24.312. (2011). 3rd Generation parnership project; technical specification group core network and terminals; access network discovery and selection function (ANDSF) management object (MO) (Release 10). Tech. Rep., 3GPP

Adamopoulou, E., Demestichas, K., Koutsorodi, A., & Theologou, M. (2005). Intelligent access network selection in heterogeneous networks - simulation results. In *proceedings of the 2nd International Symposium on Wireless Communications Systems (ISWCS 2005)* (pp. 279–283). Siena, Italy: IEEE Communications Society.

Alkhawlani, M., & Ayesh, A. (2008). Access network selection based on fuzzy logic and genetic algorithms. *Advances in Artificial Intelligence*.

Andrews, J., Ghosh, A., & Muhamed, R. (2007). *Fundamentals of WiMAX – understanding broadband wireless networking* (1st ed.). United States: Prentice Hall.

Baker, F. (2011). Views of IPv6 Site Multihoming. *The Internet Protocol Journal, 14*(2), 14–22.

Bakshi, B., Krishna, P., Vaidya, N., & Pradhan, D. (1997). Improving performance of TCP over wireless networks. In *Proceedings of the 17th IEEE international conference on distributed computing systems (ICDCS'97)* (pp. 365–373). Washington, DC, USA: IEEE Computer Society.

Balasubramanian, A., Mahajan, R., & Venkataramani, A. (2010). Augmenting mobile 3G using WiFi. In *MobiSys'10, June 15–18* (pp. 209–222). San Francisco, USA: ACM.

Bari, F., & Leung, V. (2007). Automated network selection in a heterogeneous wireless network environment. *IEEE Network, 21*(1), 34–40.

Chee, K., & David, J. (1989). Packet data transmission over mobile radio channels. *IEEE Transactions 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.

Dimatteo, S., Pan, H., Bo, H., & Li, V. (2011). Cellular traffic offloading through WiFi networks. In *Proceedings of the 8th international conference on mobile adhoc and sensor systems (MASS)* (pp. 191–201). Valencia, Spain: IEEE.

Farinacci, D., Fuller, V., Lewis, D., & Mayer, D. (2013). Locator/ID separation protocol (LIPS). Category: Experimental. <http://tools.ietf.org/html/rfc6830>.

Fekete, G. (2010). *Network interface management in mobile and multihomed nodes* (Ph.D. thesis), University of Jyvaskyla, Jyvaskyla, Finland.

Ford, A., Raiciu, C., Handley, M., & Bonaventure, O. (2012). TCP extensions for multipath operation with multiple addresses. Intended status: Experimental. <http://tools.ietf.org/html/draft-ford-mptcp-multi-addressed-09>.

Frantti, T. (2012). Expert system for open-loop power control of wireless local area networks. *Expert Systems with Applications, 39*(11), 10191–10201. Elsevier Science.

Frantti, T., & Majanen, M. (2011). Fuzzy based flow management of real-time traffic for quality of service in WLANs. Intech. (1st ed.), Croatia.

Frantti, T., & Majanen, M. (2010). *Internet traffic shaping in WLANs by packet size control* (1st ed.). NY: Nova Science Publishers, Inc..

Iannone, L., Lewis, D., Mayer, D., & Fuller, V. (2013). LIPS EID block. Intended status: Informational. <http://tools.ietf.org/html/draft-ietf-lisp-eid-block-04>.

ITU-T, (1993). ITU-T recommendation G.114. Tech. Rep., International Telecommunication Union, Geneva, Switzerland.

ITU-T. (2004). Next generation networks – frameworks and functional architecture models. General overview of NGN. Tech. Rep., International Telecommunication Union, Geneva, Switzerland.

ITU-T. (2004). Next generation networks – frameworks and functional architecture models. General principles and general reference model for next generation networks. Tech. Rep., International Telecommunication Union, Geneva, Switzerland.

Juuso, E. K. (1993). Linguistic simulation in production control. In Pooley, R., Zobel, R. (Eds.), *Proceedings of the UKSS'93 conference of the united kingdom simulation society, Keswick, UK* (pp. 34–38).

Kandula, S., Lin, K. C., Badirkhanli, T., & Katab, D. (2008). FatVAP: Aggregating AP backhaul capacity to maximize throughput. In *Proceedings of the 5th USENIX symposium on networked systems design and implementation (NSDI 2008)*, San Francisco, USA.

Kohler, E., Handley, M., & Floyd, S. (2006). Datagram congestion control protocol (DCCP). Category: Standards track. <http://tools.ietf.org/html/rfc4340>.

Kohler, E., Handley, M., & Floyd, S. (2006). Generalized connections in the datagram congestion control protocol. RFC status: Internet draft.

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

Korhonen, J., & Wang, Y. (2005). Effect of packet size on loss rate and delay in wireless links. *Proceedings of the IEEE conference on wireless communications and networking (WCNC 2005)* (vol. 3, pp. 1608–1613). New Orleans, USA: IEEE Communications Society.

Lee, S., Lee, S. (2012). User-centric offloading to WLAN in WLAN/3G vehicular networks. *Wireless personal communications*, Springer Science+Business Media.

Lettieri, P., & Srivastava, M. B. (1998). Adaptive frame length control for improve wireless link range and energy efficiency. In *Proceedings of the IEEE international conference on computer communications (INFOCOM'98)* (pp. 564–571). San Francisco, USA: IEEE Communications Society.

Mehani, O., Boreli, R., Maher, M., & Ernst, T. (2011). User- and application-centric multihomed flow management. In *Proceedings of the 36th IEEE conference on local computer networks (LCN 2011)* (pp. 26–34). Bonn, Germany: IEEE Computer Society.

Moskowitz, R., Heer, T., Jokela, P., & Henderson, T. (2012). Host identity protocol version 2 (HIPv2).

Nikander, P., Henderson, T., Vogt, C., & Arkko, J. (2008). End-host mobility and multihoming with the host identity protocol. Category: Experimental.

Nordmark, E., & Bagnulo, M. (2009). Shim6: Level 3 multihoming shim protocol for IPv6.

Ogata, K. (2009). *Modern control engineering* (5 ed.). New Jersey: Prentice Hall,.

Petander, H. (2009). Energy-aware network selection using traffic estimation. In *proceedings of the 1st ACM workshop on mobile internet through cellular networks (MICNET 2009)* (pp. 55–60). New York, USA: ACM.

Psaras, I., & Mamatas, L. (2011). On demand connectivity sharing: Queuing management and load balancing for user-provided networks. *Computer Networks, 55*(2), 399–414.

Qualcomm. (2010). 3G/Wi-Fi seamless offload. White paper. <http://www.qualcomm.com/media/documents/files/qualcomm-research-3g-wifi-seamless-offload.pdf>.

Sankarasubramaniam, Y., Akyildiz, I., & McLaughlin, S. (2003). Energy efficiency based packet size optimization in wireless sensor networks. In *Proceedings of the first IEEE international workshop on sensor network protocols and applications* (pp. 1–8). Anchorage, USA: IEEE Communications Society.

Scharf, M., & Ford, A. (2012). MPTCP application interface considerations. Intended status: Informational.

Schiller, J. (2005). Inter-AS traffic engineering case studies as requirements for IPv6 multihoming solutions.

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. *Proceedings of the IEEE global telecommunications conference (GLOBECOM'00)* (Vol. 3, pp. 1427–1431). San Francisco, USA: IEEE Communications Society.

Singh, J., Alpcan, T., Agrawal, P., & Sharma, V. (2010). A Markov decision process based flow assignment framework for heterogeneous network access. *Wireless Networks, 16*(2), 481–495.

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

Song, Q., & Jamalipour, A. (2005). Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques. *IEEE Wireless Communications, 12*(3), 42–48.

Steward, R. (Ed.), (2007). Stream control transmission protocol. Category: Standard track. http://www.rfc-editor.org/rfc/rfc4960.txt.

Steward, R., Xie, Q., Tuexen, M., Maruyama, S., & Kozuka, M. (2007). Stream control transmission protocol (SCTP) dynamic address reconfiguration. Category: Standards track, <http://tools.ietf.org/html/rfc5061>.

Steward, R., Tuexen, M., Poon, K., Lei, P., & Yasevich, V. (2011). Sockets API extensions for stream control transmission protocol. Category: Informational. http://tools.ietf.org/html/draft-ietf-tsvwg-sctpsocket-22>.

Tsirtsis, G., Soliman, H., Montavont, N., Giaretta, G., & Kuladinithi, K. (2011). Flow bindings in mobile IPv6 and network mobility (NEMO) basic support.

Wakikawa, R., Devarapalli, V., Tsirtis, G., Ernst, T., & Nagami, K. (2009). Multiple care-of addresses registration.

Wang, H. J., Katz, R. H., & Giese, J. (1999). Policy-enabled handoffs across heterogeneous wireless networks. In *Proceedings of the 2nd workshop on mobile computing systems and applications (WMCSA 1999)* (pp. 51–60). New Orleans, USA: IEEE Communications Society.

WG, I. (2009). IEEE standard for local and metropolitan area networks – Part 21: Media independent handover services.

Wilson, A. L., Lenaghan, A., & Malyan, R. (2005). Optimising wireless access network selection to maintain QoS in heterogeneous wireless environments. In *Proceedings of the 14th wireless personal multimedia communications symposium (WPMC 2005)* (pp. 1236–1240). Brest, France: IEEE Communications Society.

Xing, B., & Venkatasubramanian, N. (2005). Multi-constraint dynamic access selection in always best connected networks. In *Proceedings of the second annual international conference on mobile and ubiquitous systems: Networking and services (MobiQuitous 2005)* (pp. 56–64). San Diego, USA: IEEE Computer Society.

Yahiya, T. A., & Chaouchi, H. (2009). An optimized handover decision for heterogenous wireless networks. In *Proceedings of the 4th workshop on performance monitoring, measurement and evaluation of heterogenous wireless and wired networks (PM2HW2N 2009)* (pp. 137–142). Tenerife, Spain: ACM.

Yao, J., Kanhere, S. S., & Hassan, M. (2009). Geo-intelligent traffic scheduling for multi-homed on-board networks. In *Proceedings of the ACM international workshop on mobility in the evolving internet architecture (MobiArch 2009)*, Krakow, Poland.

Younis, M., Farrag, O., & D'Amico, W. (2009). Packet size pptimization for increased throughput in multi-level security wireless networks. In *Proceedings of the military communications conference (MILCOM 2009)* (pp. 1–7). Boston, USA: IEEE Communications Society.