Budapest University of Technology and Economics

Lightweight Control Techniques for Supporting Voice Communication over Packet Switched Networks

István Szabó
High Speed Networks Laboratory
Department of Telecommunications and Telematics

Ph. D. Dissertation

Supervisor
Dr. Sándor Molnár
High Speed Networks Laboratory
Department of Telecommunications and Telematics

Budapest, Hungary
2003

Budapesti Műszaki és Gazdaságtudományi Egyetem

Beszédátvitelt támogató jelzésmegoldások csomagkapcsolt hálózatokban

Szabó István
Távközlési és Telematikai Tanszék
Nagysebességű Hálózatok Laboratórium

Ph. D. értekezés

Tudományos vezető
Dr. Molnár Sándor
Távközlési és Telematikai Tanszék
Nagysebességű Hálózatok Laboratórium

Budapest
2003

Contents

Acknowledgements

I would like to thank my advisor Dr. Sándor Molnár for helping my initial steps as a Ph.D. student and also for guiding me through the complex and difficult task of writing a dissertation.

I am grateful to Leslie Graf of Ericsson Australia for supervising my work on AAL2 Signalling. I would also like to thank all the members of the ATLE signalling team for the suggestions and constructive criticism I received.

Szabolcs Malomsoky was always kind to review my draft publications and to comment on my ideas and research results. His help was invaluable.

The insightful comments I received from Balázs Szviatovszki and Dr. Csaba Antal helped me in finalising this dissertation.

I would like to thank Péter Lábody for the support he provided in the investigations presented in Section 3.3.5.

I had some inspiring "philosophical" discussions with Prof. Tamás Henk, Head of HSNLab, Dr. Gábor Fodor and Dr. László Hévizi about what telecommunication research is and what makes a Ph.D. Thank you.

I am grateful to Dr. Miklós Boda, Head of Research and Development at Ericsson Hungary. His commitment to create a world–class telecommunication research center in Hungary and his continuous encouragement during my work meant a lot to me.

I would like to thank the efforts and sacrifices my mother made to support my studies. I would like to thank the support I received from my wife.

List of Abbreviations

A2U	AAL2 User
AAL	ATM Adaptation Layer
AMR	Adaptive Multi–rate Speech Codec
ATM	Asynchronous Transfer Mode
BICC	Bearer Independent Call Control
B–ISUP	Broadband ISUP
BGP	Border Gateway Protocol
BS	Base Station
CAC	Call Admission Control
CBR	Constant Bit Rate
CID	Channel Identifier
DSAID	Destination Signalling Association Identifier
DSS1	Digital Subscriber Signalling System No. 1
EWMA	Exponentially Weighted Moving Average
GGSN	Gateway GPRS Support Node
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISUP	ISDN User Part
ITU	International Telecommunication Union
MTP	Message Transfer Part
NIMI	National Internet Measurement Infrastructure
NNI	Network–Network Interface
O&M	Operation and Maintenance
OSAID	Originating Signalling Association Identifier
QoS	Quality of Service
PSTN	Public Switched Telephone Network
RA	Radio Application
RED	Random Early Discard
RNC	Radio Network Controller
RTP	Real–time Transport Protocol
RTCP	Real–time Transport Control Protocol
RTT	Round Trip Time
SAAL	Signalling ATM Adaptation Layer

SGSN	Serving GPRS Support Node
SHO	Soft Handover Device
SAP	Service Access Point
SB	Signalling Bearer
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SS7	Signalling System No. 7
SSCOP	Service Specific Connection Oriented Protocol
SSCS	Service Specific Convergence Sublayer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UNI	User–Network Interface
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP
VCC	Virtual Channel Connection
W–CDMA	Wideband Code Division Multiple Access

Chapter 1

Introduction

There are three important trends in the development of telecommunication systems over the past couple of years. The first important development is the evolution of public landline mobile network infrastructure, which makes possible to offer sophisticated real–time multimedia and high speed data transfer services to mobile customers in addition to the widely used mobile telephony. This new networking infrastructure is referred as 3G or UMTS [1, 2]. Tremendous research effort has been committed to extend IP networks with the capability of offering differentiated services and providing QoS guarantees for a subset of traffic flows. Finally, these two networks are converging, IP transport network will be deployed in the fixed part of the mobile infrastructure and mobile terminals are expected to include an IP protocol stack and be capable of accessing web and real–time multimedia content over the Internet. My research was focused on control techniques, which can be used to support carrying real–time (mainly narrowband voice) traffic over packet switched networks, and it takes also into account that the above mentioned evolution and convergence results in a fast changing and quite unstable networking environment, where incremental deployment of new services and solutions is a must.

In cellular wireless systems the most precious resource is the radio spectrum. In order to effectively support packet switched services over the radio interface, a new technology, W–CDMA [3] has been developed for UMTS, which enables guaranteeing QoS over the air interface, while allocating radio resource with fine granularity. After completing W–CDMA, research has been started to build a fixed transport network infrastructure (UMTS Terrestrial Radio Access Network) around it. At that time ATM has been the most mature network technology, capable of matching W–CDMA in terms of QoS guarantees and flexibility of resource allocation. In addition, many incumbent telecommunication operators – especially in Japan, where the need for the next generation of cellular networks was most articulated – had huge capacity ATM networks, which they wanted to utilize. In order to efficiently support compressed voice communication, and to satisfy other radio access network specific requirements, ATM needed a major enhancement. A new adaptation layer has been developed, and a corresponding new switching level has been introduced. My research was focused on a new signalling protocol which makes possible the introduction of this new switching level.

In the meantime, IP took off as a promising candidate to be the base technology for a QoS enabled, multiservice network, and more and more enterprises and new telecommunication service providers wanted to carry not just data traffic but also telephony sessions over IP networks. In order to provide toll quality voice communication, the traditional best effort service paradigm is not sufficient. Some method is needed to offer prioritised treatment for real-time traffic, and block new call arrivals if the capacity limit of the network is reached. Many approaches have been proposed in the past couple of years for service differentiation and resource allocation in IP networks [4, 5, 6, 7, 8]. These methods represent different trade–off between efficient utilisation of bandwidth and implementation complexity. A significant problem hindering the widespread deployment of IP QoS mechanisms is the large installed base of equipment supporting only the best effort service. My research was focused on finding a lightweight call admission control method, which is capable of providing soft, but reasonable QoS guarantees for IP telephony, and the offered QoS guarantees are not much effected by intermediate islands of best effort routers.

1.1 Research Objectives

The general aim of my research was to find and solve open research problems with regards to efficiently carrying (compressed) voice traffic over packet switched networks.

I dealt with 3rd generation mobile radio access networks (UTRAN). The goal of my research was to propose a new signalling protocol (AAL2 Signalling), which is capable of supporting a new switching level (AAL2 switching) on top of the well–known ATM switching. Earlier, signalling solutions were tied to a specific bearer network. Further goal was to come up with a new protocol architecture, which allows AAL2 Signalling to be independent of the underlying transport network, and makes it possible to operate the protocol over IP, ATM or traditional SS7 signalling network. Finally, I had to analyse protocol implementation options to see their effect on the speed of connection establishment, which is a crucial factor when it comes to supporting soft handover in UTRAN.

One of the most important design principle of Internet was the so called "end–to–end argument"[9]. Applying this principle means that the network nodes should be bothered only with functionality, which is required by all applications, and the functionality, which is required only by a subset of end–systems shall be implemented by applications. QoS support and call admission control is definitely a functionality, which is not needed for all end–users, so applying the end–to–end argument literally routers in the network should not be burdened with call admission control. The goal of my research was to propose a call admission control solution which does not require any involvement from the routers, and to evaluate what sort of QoS guarantees such a solution can provide.

Chapter 2

A Signalling Protocol for Supporting AAL2 Switching

ATM AAL2 has been selected as the transmission technology in the radio access network of UMTS systems. To make possible deploying AAL2 in a network where support for soft handoff is essential, and to fully exploit its advantages in more traditional environments, introduction of a new switching level on top of ATM switching was found to be beneficial. I developed a new signalling protocol (AAL2 Signalling) capable of establishing, maintaining, and releasing dynamic, on–demand, end–to–end AAL2 connections. I also proposed a new protocol architecture (Bearer Independent Signalling Protocol Architecture) which makes possible easy deployment of the protocol in theoretically any network environment. When it comes to supporting soft handoff, the most important requirement is fast connection setup. I investigated the effect of introducing priority queuing of the AAL2 signalling messages on AAL2 connection setup time. My proposals have been incorporated into two new ITU–T Recommendations Q.2630.1 [10] and Q.2150.1 [11]. Q.2630.1 has later been standardised by 3GPP [12] to be used to control AAL2 connections in UMTS Terrestrial Radio Access Networks.

2.1 Background and Problem Statement

ATM Adaptation Layer 2 (AAL2) has recently been standardised by ITU-T [13]. It was designed to make possible carrying small data packets on top of an ATM infrastructure with low delay but still with efficient use of bandwidth. Bandwidth efficiency is achieved by supporting multiplexing of different sources on a single ATM virtual connection. A detailed analysis of AAL2, and comparison with other AALs shows that AAL2 is very suitable for carrying compressed voice and data having similar characteristics [14, 15].

An important consequence of the radio interface technology (Wideband CDMA) developed for UMTS is the need for soft handoff [16], when supporting mobility in the UMTS Radio Access Network. Soft handoff means that in regions, where the cells of adjacent base stations are overlapping, the mobile terminal might simultaneously be connected to more than one base station. In a nutshell, this is needed because all mobiles are interfering with each others' signals to the base stations. In order to give each mobile the quality it needs, power control is used so that each mobile signal is received in the base station at an appropriate level. When a mobile is moving away from the base station, it is continuously ordered by the base station to raise its transmission power. The increased transmission power causes more and more interference as the mobile moves closer and closer to a new base station. As soon as the mobile is connected to both base stations, its transmission power is under the control of both base stations, so the interference is alleviated. Faster connection establishment towards the new base station results in a shorter period of heavy interference, which leads to a better overall system performance.

AAL2 has been selected as the transmission technology in the landline part of the radio access network of UMTS systems. As Figure 2.1 depicts, AAL2 connections are established between the soft–handoff (SHO) device located in the serving radio network controller (RNC) node, and the Radio Application (RA) in the base stations (BS) for transferring user data destined to (originated from) a mobile station. To make it possible to use AAL2 in a network where support for soft handoff is essential, introduction of a new switching level on top of ATM switching is beneficial [14, 17]. To support AAL2 switching, a new signalling protocol (AAL2 Signalling) is required that handles on demand establishment, maintenance and tear down of end–to–end AAL type 2 connections. (E.g. an AAL type 2 connection between a Base Station (BS) and a SHO). AAL2 Signalling protocol applicable in 3rd generation mobile access networks shall meet the following requirements:

- Provide mechanisms for the establishment and clearing of point–to–point AAL2 connections over an AAL2 network comprised of AAL2 endpoints and AAL2 switching points.
- Support hop–by–hop routing.
- Be able to control AAL2 connections on more than one underlying ATM VCC that can either be permanent or switched ATM VCC.
- Be independent from the underlying signaling bearer protocol stack.
- Include a mechanism to enable seamless extension of the protocol in the future.
- Support transparent transfer of served user generated information between originating and terminating AAL2 connection endpoints.

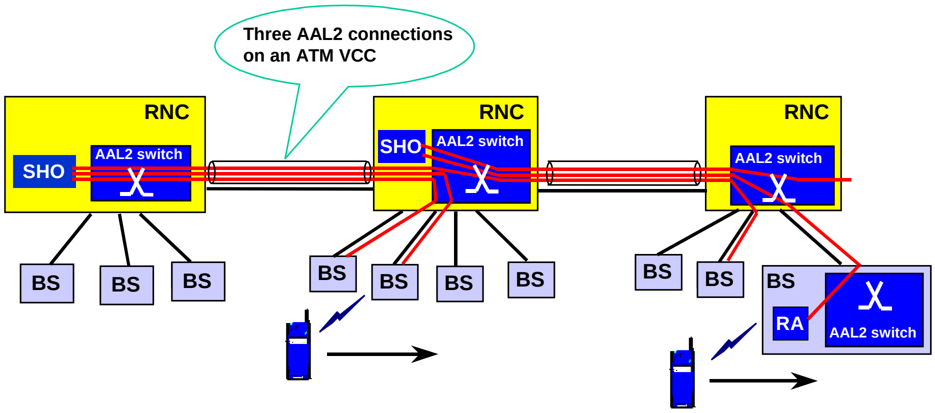

Figure 2.1: AAL type 2 connections in a UMTS radio access network

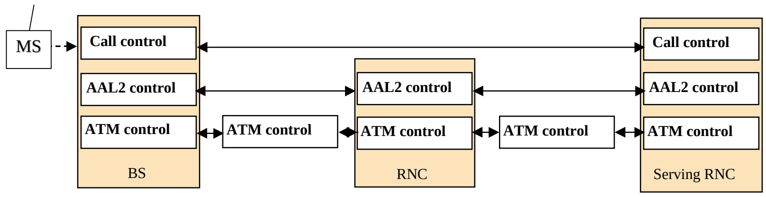

Figure 2.2: Protocol Layering in the Control Plane

- Be a symmetric protocol which means that interconnected peer protocol entities has the same capabilities for controlling AAL2 connections.

When designing the protocol that fulfills the above listed requirements the main design goal shall be to achieve very fast AAL2 connection establishment as required for supporting soft handoff. The goal of my research was to construct a signalling protocol which fulfills all the requirements listed above and to investigate techniques for optimising the connection setup delay performance of the new protocol.

2.2 AAL2 Signalling Protocol

AAL2 Signalling Protocol is able to establish and release end–to–end AAL2 connections in a network comprised of AAL2 endpoints and AAL2 switches. The call handling is done in the layers above the AAL2. Not burdening the AAL2 signalling with call control information makes the protocol simple and fast. This is very crucial when supporting soft handoff and enables the new protocol to be applied in different application scenarios as a connection control protocol which operates under the supervision of an application scenario specific call control protocol (Figure 2.2)

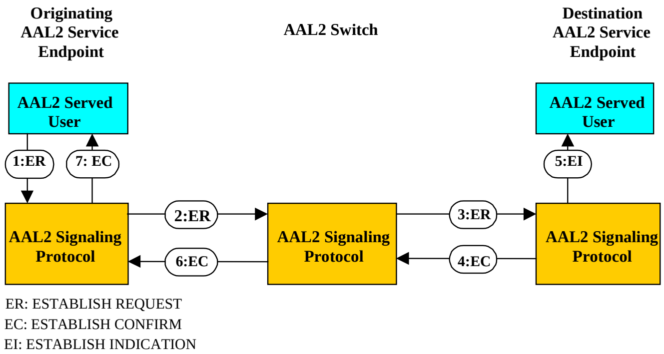

Figure 2.3: Sequence of primitives and protocol messages in case of successful AAL2 connection establishment

Evidently, the most important functionality of the AAL2 Signalling protocol is the establishment and the release of point–to–point AAL2 connections as requested by AAL2 Served Users. The AAL2 Served User in UMTS networks is the radio resource management and handover control entity, which requests AAL2 connections when new soft handover legs are established.

Figure 2.3 depicts the sequence of primitives and protocol messages used for establishing a new AAL2 connection. It is noteworthy that the AAL2 Signalling Protocol at the destination Service Endpoint acknowledges the connection establishment (Step 4: ESTABLISH CONFIRM), and then informs the served user about the new incoming connection (Step 5: ESTABLISH INDICATION). This method contributes to a low and predictable AAL2 connection establishment delay. Not asking the AAL2 served user before accepting the connection does not cause problems because served users can properly be co–ordinated by upper layer functionality. E.g. such functionality is the radio resource management and handoff control entity in UMTS networks.

In order to establish an end–to–end AAL2 connection, the following information needs to be carried in the ESTABLISH REQUEST message:

- AAL type 2 Service Endpoint Address to identify the destination endpoint.
- An ATM VCC identifier to refer to the ATM VCC which will carry the AAL2 connection.
- A CID value to unambiguously identify the AAL2 connection inside the ATM VCC.
- Link Characteristics, or SSCS Information to describe the AAL2 flow.

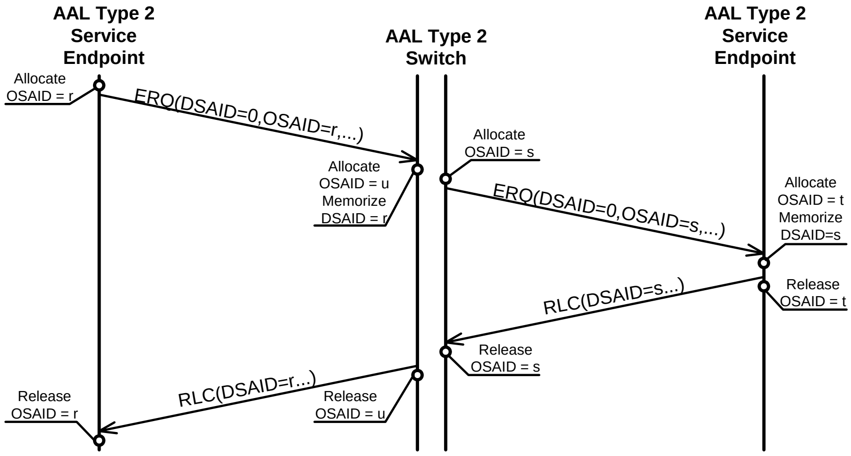

Figure 2.4: Unsuccessful connection establishment — example for handling of Signalling Association Identifiers

The protocol carries a parameter called Served User Generated Reference in the ESTABLISH REQUEST message. In a UMTS base station, the AAL2 Served User uses this information to map an incoming AAL2 connection to the corresponding radio channel, which carries the information further towards the Mobile Terminal.

ESTABLISH REQUEST messages optionally include a Served User Transport parameter, which can be used to carry maximum 255 octets of information transparently through the AAL2 switching network. If the connection being established is for test purposes, it is indicated by a flag in the ESTABLISH REQUEST message.

AAL2 connections are released by a simple and straightforward link by link procedure.

To further enhance the speed of the connection establishment procedure, the protocol messages and parameters are coded to allow fast processing. For example, a single coding standard is established, and each message carries an identifier that makes it possible to distribute the incoming message to the corresponding protocol machine without going through internal translation tables. This mechanism is shown in Figure 2.4 with the example of message sequence for unsuccessful connection establishment.

Signalling Association Identifiers (Originating SAID & Destination SAID) are used to associate an incoming signalling message with a particular connection instance. In order to allow an incoming signalling message to find its way to the corresponding protocol entity without going through a translation table, SAIDs are treated in the following way:

1. Whenever a new signalling procedure is started, a new protocol entity instance is created and an OSAID is allocated to it; this ID is then

transported in the first message in the OSAID parameter. The DSAID in this message contains the value "unknown", i.e., all octets are set to "0". (In the Figures, this is indicated by "DSAID=0".)

2. Upon receipt of a message that has a DSAID field set to "unknown", it is perceived as starting of a new signalling procedure, therefore a new protocol entity instance is created and an OSAID is allocated to it.

3. In the first message returned to the initiator of the signalling procedure, the OSAID of the sending protocol entity instance is transported. The DSAID field carries the previously received OSAID of the initiator of the procedure.

4. In all subsequent messages, the DSAID field carries the previously received OSAID of the destination entity.

5. If the first message returned to the initiator of the signalling procedure is also the last one for this procedure, no OSAID parameter is carried in the message. The DSAID field carries the previously received OSAID of the initiator of the procedure.

AAL2 Signalling Protocol provides basic maintenance capabilities to cope with congestion in the signalling network and temporary outage of signalling links and of peer switching entities. It is able to reduce or cut the connection requests generated towards congested AAL2 switches.

AAL2 Signalling offers a Reset mechanism to support consistent recovery from fault cases in AAL2 switches. The reset procedure is invoked under abnormal conditions such as when the current status of the channels is unknown or ambiguous, for example, an AAL type 2 switching system that has suffered memory mutilation will not know the status of channels in one or several AAL type 2 paths. All the affected channels and any associated resources (e.g., bandwidth, etc) between the two adjacent AAL type 2 nodes shall be released. The resources are made available for new traffic. The reset procedure covers the following three cases:

1. Reset all channels used for user plane traffic in all the AAL type 2 paths between two peer AAL type 2 nodes.

2. Reset all channels used for user plane traffic in a single ATM VCC between two peer AAL type 2 nodes.

3. Reset a single channel between two peer AAL type 2 nodes.

AAL2 Signalling protocol contains a mechanism for Blocking and Unblocking resources. The AAL2 blocking procedure is provided to prevent an ATM VCC from being selected for carrying new connections other than test connections. Existing connections on the AAL type 2 path are not affected. This feature can be used during test procedures (Blocking out user plane traffic from an ATM VCC while testing it.), maintenance procedures (Blocking out an ATM VCC while changing the hardware board handling its traffic.), and can also be used to control the load in the control plane as well as in the user plane (Reducing the number of ATM VCCs available for user plane traffic reduces the number of AAL2 connections that need to be handled.).

Blocking can be initiated by either signalling endpoint. When blocking is invoked, both ends of the ATM VCC are put into blocked state. A blocked ATM VCC cannot be selected for new, non–test traffic by the AAL type 2 nodes. An acknowledgment is required for each blocking and unblocking request. Unblocking can only be initiated by the same AAL type 2 node which initiated the blocking procedure. It is performed by sending an unblocking request. At either end, the blocked state is removed and the ATM VCC is made available again for all new connections.

A forward and backward compatibility mechanism is included in the protocol that enables extending the protocol in the future without causing problems for switches operating according to an earlier version of the protocol. The mechanism is based on instruction indicators included in signalling messages and parameters. The indicators prescribe what a switch should do if it receives unrecognised information. Possible actions are:

- Proceed with the connection establishment and pass the unrecognized information transparently on to the succeeding switch.
- Proceed with the connection establishment but discard the unrecognized information. Sending a notification to the initiator of the unrecognized information may also be requested.
- Abandon the connection establishment and send a notification to the originator of the unrecognized information.

This simple mechanism allows that services which are not yet supported by all nodes in the network can work, provided that they only require transparent transfer of information by the intermediate nodes.

AAL2 signalling protocol is a separate new protocol and not an extension of one of the ATM signalling protocols. The benefit of this approach is that switched AAL2 connections can be established on top of any ATM network regardless of the protocol used for establishing ATM level connections. The ATM level connections can be established using any existing ATM signalling protocol (e.g.: ITU–T B–ISUP, ATM Forum PNNI, ITU–T Q.2931, ATM Forum UNI) but (soft) permanent virtual connections can also be used at the ATM level. Further benefit of this approach is that intermediate pure ATM

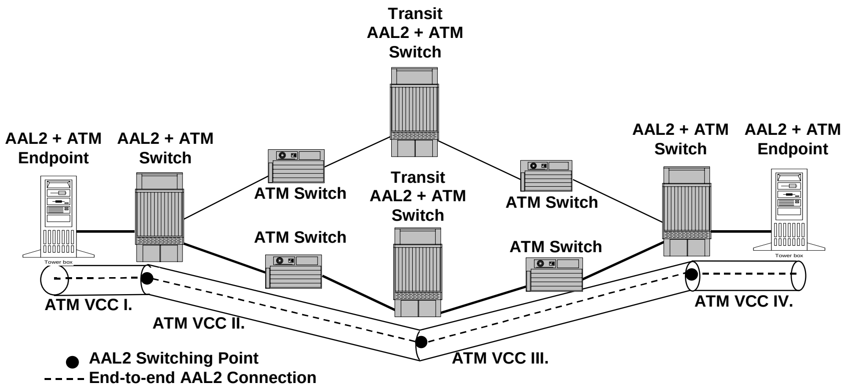

Figure 2.5: ATM virtual connections and an end–to–end AAL2 connection in an AAL2 overlay network

switches are not burdened with storing and forwarding AAL2 signalling messages. Figure 2.5 shows an AAL2 network overlying an ATM network.

An important consequence of the separation from ATM signalling is that the AAL2 network can have its own addressing plan and routing, which allows multiple AAL2 overlay networks to operate on top of an ATM network. The addressing can be changed in any of the networks without worsening the efficiency of routing in the others. This mutual independence becomes crucial when the operator of the ATM and AAL2 network is not the same organisation since it allows both party to be in charge of the addressing in its own network. (An example can be a 3^{rd} generation mobile network operator that leases the underlying ATM connections from an ATM service provider.)

AAL2 is designed to exploit the QoS, traffic management and traffic characterisation capabilities of the underlying ATM level. In order to facilitate efficient use of ATM level resources and to enable routing of AAL2 connections through a network, a limited set of AAL2 traffic parameters are proposed including Peak Bit Rate, Average Bit Rate, Peak Packet Size, Average Packet Size. Transferring voice and soft handoff branches are delay sensitive services, therefore a single, delay sensitive QoS class will be supported in the first release of AAL2 Signalling. With the help of the forward compatibility mechanism, new QoS classes can be added later if the need for them is justified.

2.2.1 Validation of AAL2 Signalling Protocol

The new protocol which I have conceptualised went through a number of filters which assure its correctness:

- I contributed with the details of the new protocol to Study Group

11 of the International Telecommunication Union [S1]–[S24]. These proposal have been rigorously checked by other members of the Study Group before incorporating them into an ITU–T recommendation.

- After completing the informal specification of AAL2 Signalling, a formal SDL description has been created by ITU–T which is documented in ITU–T recommendation Q.2630.2.

- ETSI, the European Telecommunications Standards Institute has launched a project to define a standardised test suite for AAL2 Signalling. This test suite is documented in [18, 19, 20]. During such a project, it is usual to notify the standardisation body which produced the recommendation if any inconsistency is found in the specification during test development. In this case there was no such trouble report generated.

- The AAL2 signalling protocol has been implemented by many companies (Ericsson [21], Trillium [22], Spirent Communications [23]), and it is now part of commercial product offerings.

- The Ericsson implementation has been tested using the ETSI test suite, and the implementation passed the test. [24]

As seen from the above list, the informal AAL2 signalling protocol specification has been investigated by a couple of groups from many different perspectives using a couple of formal methodologies. During these investigations no inconsistency has been discovered in the protocol specification so it can safely be considered correct.

2.3 AAL2 Signalling Protocol Architecture

AAL2 can be used in many different application scenarios, and AAL2 signalling should be designed to enable easy deployment of the new protocol in all those networks. The question that needs to be answered is which the ideal signalling bearer protocol stack is for carrying AAL2 signalling messages.

If we consider only the needs that can be deducted from the capabilities supported by the AAL2 signalling protocol, the reasonable choice is to follow ATM Forum PNNI's approach and connect the adjacent AAL2 switches with AAL5 based Signalling AAL links (ITU-T Q.2110 & Q.2130).

If the UMTS Terrestrial Radio Access Network case is considered, the broadband SS7 network (that is indispensable for carrying the mobile network related signalling) can be reused also for carrying the AAL2 signalling. Reusing the SS7 network simplifies the application of AAL2, since it eliminates the need for maintaining a separate signalling network for the AAL2.

In UMTS core networks and in landline trunking (voice over ATM) application, the legacy system used for carrying signalling messages is the narrowband SS7 network, therefore carrying the AAL2 signalling on narrowband SS7 is also beneficial.

Considering the ongoing effort in IETF with the aim of transporting SS7 signalling on IP, design of a future proof AAL2 signalling should mean that IP is also supported as a signalling bearer.

The above listed arguments leads us to the conclusion that the notion of “bearer independent signalling protocol” should be introduced and the AAL2 signalling protocol should be designed in a way that it can theoretically be carried on any available signalling bearer protocol stack.

Describing the concept at a high level, a set of “generic primitives” is defined. The signalling protocol relies on these “generic primitives” when exchanging signalling messages with the peer signalling entities, and when receiving information about the conditions in the signalling network. These generic primitives are provided by so called “bearer converters”. The responsibility of a bearer converter is to translate the generic primitives into primitives offered by a concrete signalling bearer protocol and vice versa. Depending on the generic primitives and the particular signalling bearer, a bearer converter can be a simple parameter converter but it can also be a fairly complex finite state machine.

The bearer converter hides all differences between signalling bearer services from the signalling protocol. It can monitor the availability of remote signalling entities and the signalling links in the network. The bearer converter can implement segmentation and reassembling or a sequence numbering and retransmission service depending on the requirements imposed by the underlying layer that is intended to be used as signalling bearer. The resulting protocol architecture for the AAL2 signalling protocol is depicted in Figure 2.6.

There are two fundamentally different classes of underlying signalling bearer services: connection–less and connection oriented services. As an example, SAAL–UNI is an assured mode, connection oriented data transmission capability, while MTP offers a sequenced, connection–less service. The generic primitives offered by the bearer converter can either provide a connection–less or a connection oriented service. This means that a bearer converter offering a connection–less service but operating on top of a connection oriented bearer need to take care of maintaining the signalling bearer connections that requires a simple finite state machine.

An AAL2 switch usually has multiple neighbours to communicate with. Certain signalling bearer services (e.g.: SSCOP) are link level services while others (e.g.: IP) are network level services. To avoid the need for address translation in case of operating on top of a network level bearer service, the selected approach is to create a bearer converter instance per each adjacent AAL2 switch.

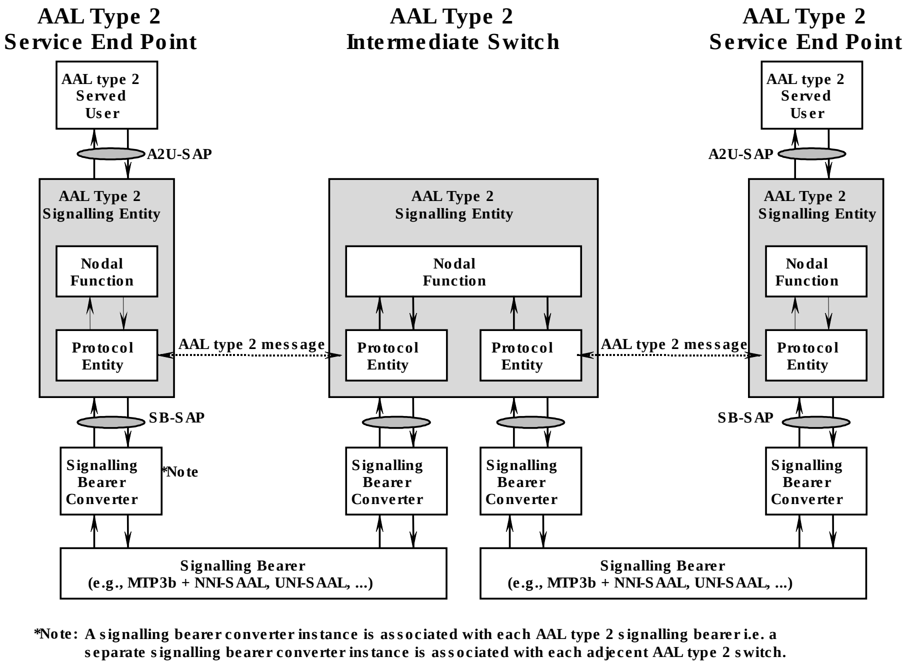

Figure 2.6: AAL2 Signalling protocol architecture

2.3.1 Generic Primitive Interface

Given the functionality of AAL2 Signalling (see Section 2.2) the generic primitive interface shall have the functionality to send and receive signalling messages, to indicate availability of the service and notify about congestion situations. AAL2 Signalling Protocol relies on the following generic primitives:

IN-SERVICE.indication primitive indicates that the signalling transport is able to exchange signalling messages with the peer entity.

OUT-OF-SERVICE.indication primitive indicates that the signalling transport is unable to exchange signalling messages with the peer entity.

TRANSFER.request conveys a signalling message from the AAL2 signalling protocol to its peer protocol entity.

TRANSFER.indication provides a signalling message from the peer entity to the AAL type 2 signalling protocol.

CONGESTION.indication is an optional primitive used to convey information concerning signalling network congestion.

Description of the bearer converters designed for adopting the AAL2 signalling protocol to the broadband version of Message Transfer Part and Signalling ATM Adaptation Layer at the UNI are presented in details in the next 2 sections.

2.3.2 Bearer Converter for the Message Transfer Part

Message Transfer Part (MTP) is a connection–less service for transporting signalling messages. Moreover, MTP provides means for reporting availability of remote signalling entities and signalling network nodes, and it is capable of indicating signalling network congestion. [25]

Routing of signalling messages relies on routing information provided by the MTP user. Before handing over an AAL2 signalling message to MTP, the signalling bearer converter shall attach the appropriate MTP routing label to the message. When receiving an incoming message, the bearer converter shall discard the MTP specific information and forward only the AAL2 signalling message to the AAL2 signalling protocol.

If MTP provides information about the unavailability of a signalling network node or that of a peer AAL2 signalling entity, both need to be translated into the generic primitive: OUT-OF-SERVICE.indication. If the peer AAL2 signalling entity is not accessible, MTP does not monitor its availability. In order to detect when the peer entity becomes available again, the bearer converter implements the following simple procedure: the bearer converter repetitively issues a message (AAL2 SIGNALLING ENTITY TEST) towards the peer signalling bearer converter entity until receiving a positive response (AAL2 SIGNALLING ENTITY AVAILABLE).

If the bearer converter detects the availability of a peer AAL2 signalling entity, or MTP provides information about the availability of a signalling network node, these indications are forwarded to the AAL2 signalling protocol in the generic primitive: IN-SERVICE.indication.

Signalling bearer converter keeps track of the information provided by the MTP with regards to signalling network congestion. When the bearer converter receives the first congestion indication from MTP, a CONGESTION.indication primitive is issued towards AAL2 signalling protocol indicating the smallest possible congestion level. At the same time, two timers "Timer_Short" and "Timer_Long" are started. During "Timer_Short" period, all received congestion indications for the same signalling network node are ignored in order not to reduce signalling traffic too rapidly. Reception of a congestion indication after the expiry of "Timer_Short", but still during period "Timer_Long", will result in a CONGESTION.indication primitive indicating that congestion level has increased by 1, and restart timers "Timer_Short" and "Timer_Long". This step wise increase of congestion level is continued until maximum value is obtained by arriving at the last step. If "Timer_Long" expires (i.e. no congestion indications having been

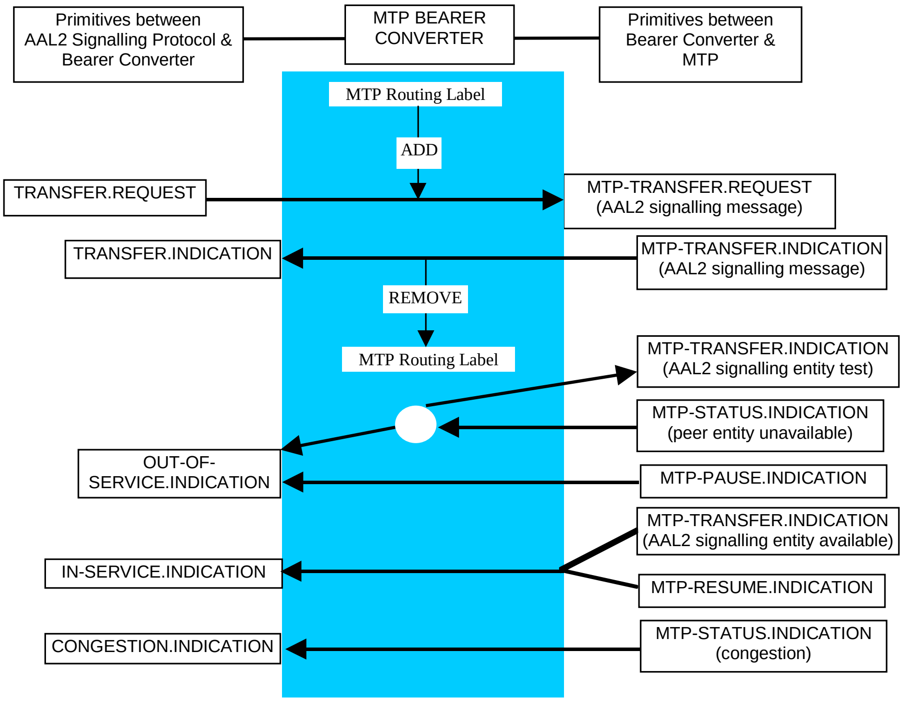

Figure 2.7: Primitive mapping in the MTP Bearer Converter

received during the "Timer_Long" period), a CONGESTION.indication reporting the congestion level decreased by 1 is issued, and timer "Timer_Long" is restarted, unless minimum value of congestion level parameter has been reached. The operation of the bearer converter is shown in Figure 2.7.

2.3.3 Bearer Converter for the Signalling ATM Adaptation Layer at the UNI

Signalling ATM Adaptation Layer (SAAL) at the UNI [26] is a connection oriented service, therefore the bearer converter, which supports the generic primitives described in Section 2.3.1, shall maintain a SAAL connection when accepting signalling messages from the upper layer. In system start–up phase, the bearer converter establishes the SAAL connection. During regular operation, the bearer converter is continuously monitoring the availability of the underlying connection. Figure 2.8 depicts the state machine required in the bearer converter to hide the connection–oriented nature of the underlying service from the AAL2 Signalling Protocol.

After successful establishment of the SAAL connection, the bearer converter sends an IN-SERVICE.indication primitive to the AAL2 Signalling Protocol, which then starts sending and receiving signalling messages.

Data transfer primitives required at the generic primitive interface match

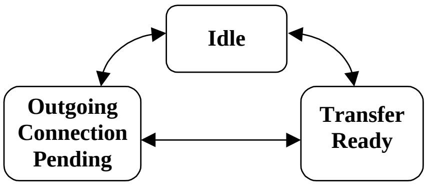

Figure 2.8: State Machine Controlling the Connection Establishment

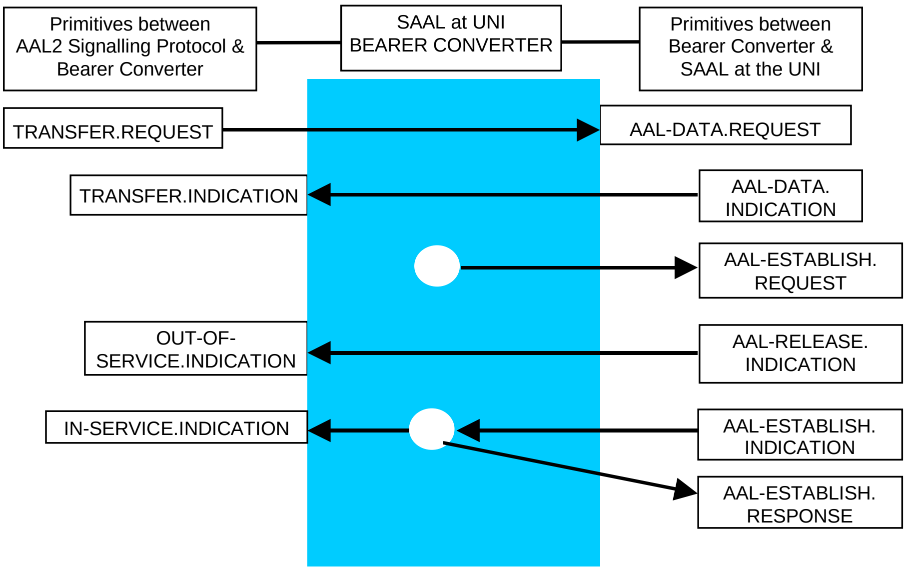

Figure 2.9: Primitive mapping in the SAAL at the UNI Bearer Converter

completely the data transfer primitives offered by the SAAL, therefore the bearer converter does not perform any data processing, just converts primitive names.

If the bearer converter detects any failure of the SAAL connection, it issues an OUT-OF-SERVICE.indication to the upper layer. After succeeding in re–establishment of the connection, an IN-SERVICE.indication is invoked. The operation of the bearer converter is shown in Figure 2.9.

Basically, the same bearer converter functionality is required to run the AAL2 Signalling Protocol on top of SSCOP. This protocol stack, combined with the assured data transfer capability of AAL type 2, makes possible a new networking arrangement for carrying signalling messages: a signalling network can be built based on AAL2 channels. In this case, there is no need for the MTP signalling network or separate AAL5 ATM VCCs for the AAL2 signalling, but the signalling can be transferred in a single AAL2 ATM VCC, sharing it with the user plane traffic.

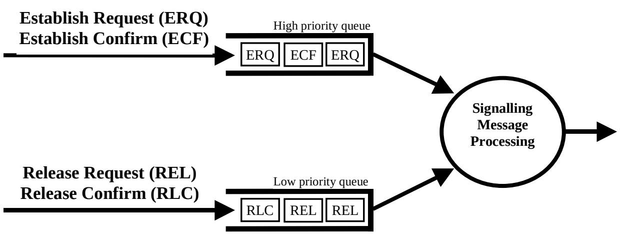

Figure 2.10: Priority Queuing of AAL2 Signalling Messages at one process

2.4 Performance Optimisation of AAL2 Signalling in UMTS Terrestrial Radio Access Networks

As stated in Section 2.1, a key performance aspect of UMTS Terrestrial Radio Access Networks is the speed of AAL2 connection establishment. A way to increase the speed of AAL2 connection establishment is to introduce prioritised handling of signalling messages, particularly, to assign higher priority to the messages involved in the connection establishment. The messages that should get higher priority are the ESTABLISH REQUEST and ESTABLISH CONFIRM message. This priority mechanism is depicted in Figure 2.10, where it is shown that two queues are implemented in all the processing elements of the AAL2 switch. The first queue collects the ESTABLISH REQUEST (ERQ) and ESTABLISH CONFIRM (ECF) messages. The messages in this queue get absolute priority over messages in the second queue, which means that the server starts fetching messages from the second queue only if the first one is empty. The second queue collects the messages that are used for connection release[1].

I investigated the effect of priority queuing on system performance using simulations and analytical calculations in single node configurations and in more realistic network settings.

2.4.1 Single Node Case

An AAL2 switch can be modelled as an M/G/1 queuing system. We can assume that the average service time of the connection setup and connection release related messages is equal. (The best approximation is probably to assume that it takes constant time to process a signalling message.) If there is no distinction between signalling messages related to connection setup

[1]Some maintenance related messages are also defined in AAL2 Signalling. The reason for ignoring these messages in this study is that these message are related to exceptional cases, and their rate is very small compared to the rate of traffic handling messages. Furthermore, it is very likely that in a real implementation such messages are not processed by the same processor that handles the traffic related messages, but rather by a dedicated processor, which is responsible for the O&M functionality of the node.

and connection release we know from elementary queuing theory [27] that the average waiting time of a signalling message can be calculated as:

$$W = \frac{W_0}{(1-\rho)} \tag{2.1}$$

where W_0 is equal to the expected time that a newly arriving message spends in the queue while the processing of the message which is serviced upon its arrival is finished. ρ is the utilisation factor.

In case of priority queuing, we can use equation 2.1 to calculate the average waiting time of the messages in the high priority class since we have strict priority queuing so the messages in the lower class have no effect on the processing of the high priority class. An ESTABLISH REQUEST message is later followed by a RELEASE REQUEST and an ESTABLISH CONFIRM is later followed by a RELEASE CONFIRM, which means that if the cumulative arrival rate of the AAL2 signalling messages is denoted by λ, the arrival rate of the messages in the high priority class λ_H and the arrival rate of messages in the low priority class λ_L is equal:

$$\lambda_H = \lambda_L = \lambda/2$$

Dividing the average waiting time of the connection establishment messages in the priority case with the average waiting time of the messages in the non–priority case we obtain a measure of the decrease in connection establishment time resulting from the priority queuing. This quotient can be written as:

$$\frac{1-\rho}{2-\rho} \tag{2.2}$$

We define the gain resulting from the introduction of priority queuing as the decrease in percentage that can be measured in the average waiting time of the connection establishment related messages in the two cases.

The average waiting time of the messages in the low priority class can be obtained using the Conservation Law [27]. This law expresses the fact that in a real system preferential treatment to a particular class of customers can only be afforded at the expense of other customers:

$$\sum_{p=1}^{P} \lambda_p W_p = \frac{\lambda W_0}{1-\rho} \tag{2.3}$$

where P is the number of traffic classes (in our case $P = 2$), λ_p is the arrival rate of messages in class p, and W_p is the average waiting time of class p messages.

Solving the set of linear equations resulting from 2.3, the average waiting time in the low priority class (W_L) is:

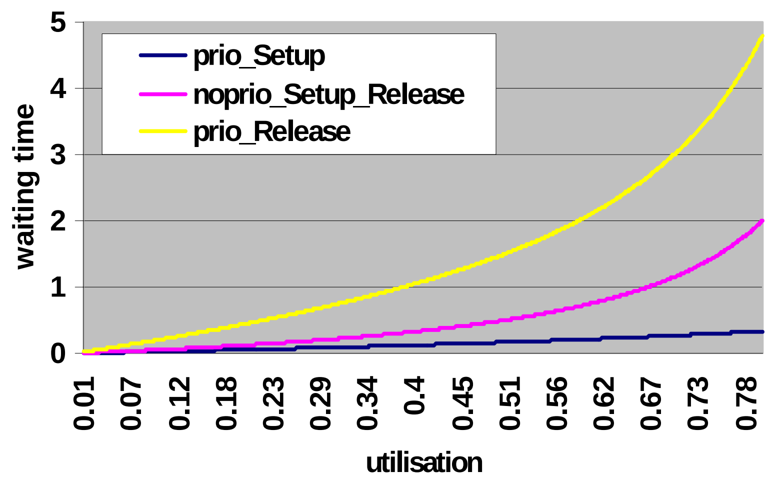

Figure 2.11: Evolution of the Average Waiting Time of the Setup and Release Message Types

$$W_L = \frac{2W_0}{(1-\rho)(2-\rho)} \tag{2.4}$$

Figure 2.11 depicts the average waiting time of the signalling messages as the processor utilisation increases, while Figure 2.12 shows the percentage gain as defined above.

2.4.2 AAL2 Switching Network Case

The UTRAN transport network has some salient architectural constraints, which are taken into account in the investigation. In a UTRAN network, AAL2 switches are located in radio network controllers (RNC) and base station (BS) nodes. A BS is connected to its controlling RNC, and there is no direct link between base stations. Furthermore, AAL2 connections are always initiated by the SHO unit and terminated in the BS. In theory, it is possible to establish AAL2 connections from an SHO located in a particular RNC, to any BS in the network when tracking a fast moving mobile. However, the so called SHO–relocation is specified in the UTRAN standards [28], which limits the AAL2 connections to span a maximum of 2 RNCs, in order to save transmission resources, and to decrease the delay of connection establishment and user data. Based on these considerations, I constructed two sample networks for the simulations, which are typical in a UMTS Terrestrial Radio Access Network. In the first case (Figure 2.13), 2 RNCs are interconnected by a link, and 10 base stations are controlled by each RNC. The base stations are connected in a tree topology. In the second case, the difference is that all the 10 base stations are directly connected to

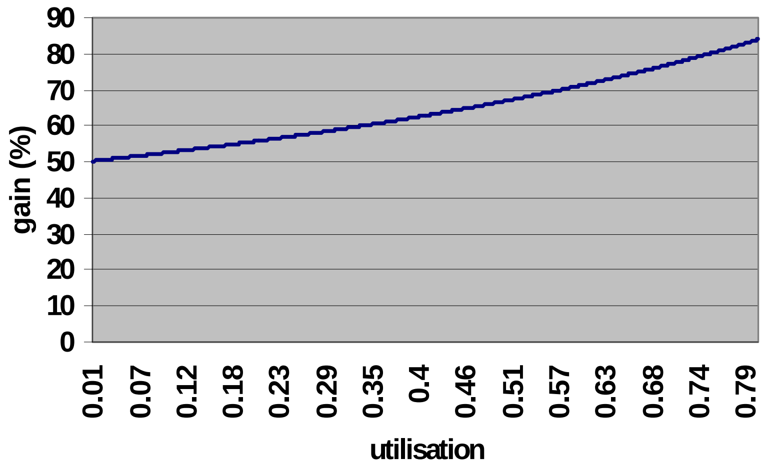

Figure 2.12: The Effect of Prioritisation on the Connection Setup Delay as the Processor Utilisation Increases

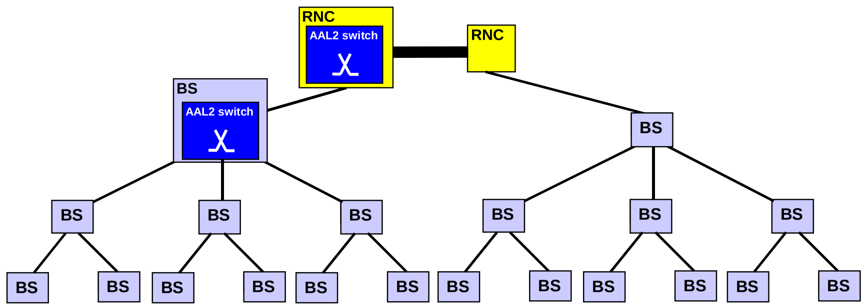

Figure 2.13: "Tree" AAL2 Switching Network Configuration

the controlling RNC. Each RNC, and BS site contains an AAL2 switching node in both configurations.

Unfortunately, these switching network cases can not be handled analytically. I have carried out simulation experiments for both network configurations, with two different node models. In the first case, the network is constructed from AAL2 nodes which apply FIFO queuing of the signalling messages. In the second case, the switching nodes give higher priority for the ERQ and ECF messages. I assumed a properly dimensioned radio access network, which means that AAL2 connection blocking does not occur due to link capacity limitations[2]. The goal of the simulation was to evaluate the effect of prioritised signalling message handling in the two network configurations. I gradually increased the AAL2 connection arrival rate, until connection requests were rejected because of signalling buffer overflow in the

[2]This is a realistic assumption, since the bottleneck should be the expensive air interface capacity in all radio access networks.

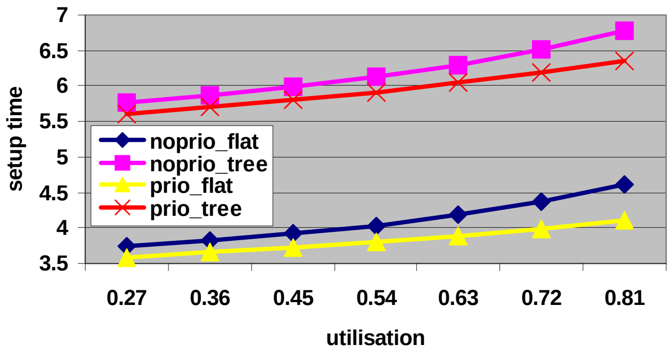

Figure 2.14: Average Connection Establishment Time Using Different Topologies

AAL2 switches. All AAL2 connections were originated by the RNCs, and terminated in one of the BS nodes. Approximately half of the connections were initiated by each RNC, and the destination endpoint was uniformly distributed between all 20 BS nodes. Figure 2.14 depicts the average time needed to set up a new AAL2 connection between an RNC and a BS when the utilisation factor of the signalling processor in the RNCs were increased from 0.27 to 0.81. The connection arrival process is Poisson, and the holding time distribution is exponential. We stopped the simulation each time after the completion of 10000 connection establishments and releases.

The average connection establishment time increases by a factor of 1.5 in the "Tree" configuration, and this increase is more or less independent of the connection arrival rate. Introducing prioritisation of the connection establishment related messages results in a significant decrease of the connection setup delay. The gain is more significant in the "Flat" network, and it grows in both network configurations as the connection arrival rate increases. Figure 2.15 compares the connection setup time improvement at different utilisation levels, and shows that the gain grows at a higher pace in the "Flat" network. Of course in a real system nothing can be bought for free. The better the connection setup performance the worse the connection release performance. However, as explained in the introduction, the most important design concern in a radio access network is the efficient use of the scarce radio resource, which is best supported by providing the fastest possible soft handoff leg establishment.

2.5 Conclusions

The work presented in the chapter is centered around building a signalling protocol which allows the introduction of a new switching level on top of ATM. I shortly discussed the UMTS Terrestrial Radio Access Network as

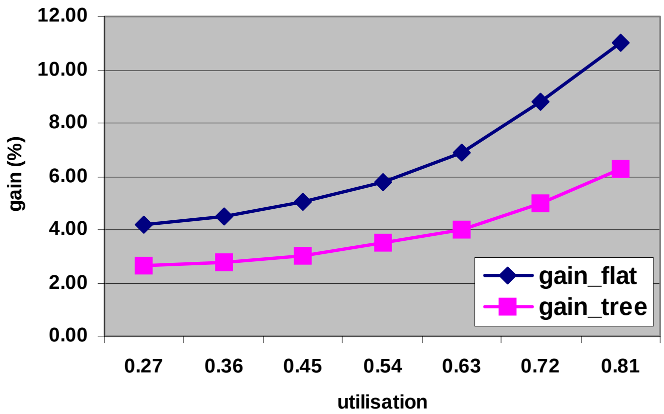

Figure 2.15: Performance of priority Queuing in the “Flat” and “Tree” Topologies

one of the most important application area of AAL2. It was indicated that due to the W-CDMA air interface technology, one of the most crucial performance parameter is the speed of connection establishment for soft handover. My research contribution includes detailed design of the new protocol called AAL2 signalling together with its evolutionary protocol architecture, which enables its use on top of many underlying network technologies. My proposals have been incorporated into new ITU–T Recommendation Q.2630.1 [10] and Q.2150.1 [11]. Q.2630.1 has later been standardised by 3GPP [12] to be used to control AAL2 connections in UMTS Terrestrial Radio Access Networks. Finally, it was demonstrated by simulation and analytical evaluation that the performance of the new protocol can further be optimised by implementing prioritised handling of different classes of protocol messages.

Chapter 3

A Core–stateless Call Admission Control Method for Supporting IP Telephony

One of the most actively studied research area of the past couple of years is the design of connection admission control methods for IP networks with different complexity, and the evaluation of the trade–off between the complexity of the method and the offered performance guarantees. A lightweight, end–to–end measurement based call admission control method is proposed here, which is tailored to the needs of IP telephony gateways. Instead of probing the network on a per–call basis, the solution relies on measuring the quality experienced by ongoing live sessions. This leads to very fast admission decision compared to existing methods, and also eliminates the load of probe packets, which is a drawback of other end–to–end measurement based solutions.

3.1 Background and Problem Statement

Both incumbent telecommunication operators and emerging new mobile and fixed telephony service providers are willing to offer telephony services over their IP network. Those IP networks are in most cases equipped with traditional best effort routers, and it will take significant amount of time and money to upgrade all of them with Differentiated Services [5] features. In order to provide acceptable quality voice communication, the best effort service paradigm is not sufficient. Some method is needed to block new call arrivals if the capacity limit of the network is reached. Many approaches have been proposed in the past couple of years for service differentiation and resource allocation in IP networks. These methods represent different trade–off between efficient utilisation of bandwidth and implementation complexity.

The application scenario is depicted in Figure 3.1. The access network

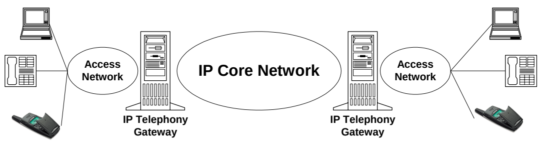

Figure 3.1: Overview of the Network Architecture

can be of basically any kind: it can be a traditional PSTN network or a UMTS Terrestrial Radio Access Network. It is further assumed that some sort of end–to–end call/session level signaling protocol (H.323, SIP, DSS1, ISUP, BICC, or their appropriate combination) is used to control the calls. When a call establishment message hits the gateway, the gateway makes a call admission control decision to check that the IP network has enough capacity to accommodate the new call. If the decision is positive, the call/session level signalling proceeds towards the remote end, otherwise the call is blocked.

Traditional resource provisioning methods rely on some sort of signalling protocol [6, 29] to indicate resource requests to the routers in the core of the network. Load Control [6] is a prime example of a lightweight resource control signalling solution. An IP telephony gateway applying Load control would send a test packet into the network upon arrival of each individual voice call. Load Control compliant routers in the core of the network maintain information about the aggregate traffic load. When a router receives a probe packet and it is congested, it marks the test packet. The remote gateway encapsulates the header of the test packet into the payload of a new test packet, and sends it back to the initiator. When the reverse packet arrives back to the initiator, it can determine whether the network is congested. If the network is found to be congested, the IP telephony gateway rejects the new telephony call. The resource request travels back and forth between the telephony gateways through the IP core network, which delays the connection establishment at least by a round trip time, and in order for the mechanism to work correctly each core router needs to be aware of it.

The deployment of the recently proposed end–to–end measurement based admission control methods [7, 30] is another option. These solutions can assure certain service guarantees for sessions, while their implementation complexity and deployment cost is very limited, since they do not require any upgrade of the routers in the network core.

Elek et al. [7] proposes a call admission control method, where the host probes the network before sending data, and uses forward error correction to compensate the effect of frame errors. The rate of the probe packets is equal to the peak rate of the actual data traffic. During the probe interval the destination host measures the packet loss rate, and reports it back to the initiator. If the reported packet loss ratio is below a threshold, the new session is admitted to the network.

Bianchi et al. [30] introduces an approximate analytical model to evaluate the performance of end–to–end measurement based connection admission control mechanisms. The mechanism requires senders to submit a succession of probe packets, and they are allowed to switch to data transmission phase only if the rate of the probe packets measured at the receiver is above a certain threshold. The model is reasonably accurate for constant bit rate connections, but much less accurate for variable bit rate connections. Unfortunately, the majority of modern voice and video coders generate variable bit rate traffic.

The probe stream based method has three significant drawbacks:

- The extra load caused by sending probe packets at the peak rate of the source.
- The additional delay incurred by the probe period.
- The possibility of network underutilization in case of high offered traffic due to the fact that the probe load is so high that each probing session experiences packet loss above the preset threshold, so none of them will finally be admitted.

One of the most important design principle of Internet was the so called "end–to–end argument" [9]. Applying this principle means that the network nodes should be bothered only with functionality, which is required by all applications, and the functionality, which is required only by a subset of end–systems shall be implemented by applications. QoS support and call admission control is definitely a functionality, which is not needed for all end–users, so applying the end–to–end argument literally routers in the network should not be burdened with call admission control. The goal of my research was to propose a call admission control solution which does not require any involvement from the routers, and to evaluate what sort of QoS guarantees such a solution can provide.

3.1.1 A Classification of Measurement Based Admission Control Methods

To place the method I propose into a broader perspective, this introductory section ends with a classification of measurement based admission control techniques.

Early approaches for connection admission control relied solely on a set of traffic descriptor parameters provided by the traffic source at the time of connection request. In this scenario, the connection establishment request travels through the network. Based on the traffic descriptors, each and every node calculates the resource requirement of the new connection and also maintains a variable storing the resource requirement of all previously admitted connections. The new connection is admitted if the capacity required by the existing and the new connections is below the link capacity [31]. Such method is capable of providing hard, mathematically provable QoS guarantees for each and every packet of the admitted connections.

Providing precise traffic descriptors a priori proves to be a daunting task. On the other hand, if the traffic descriptors are not tight enough, network resources are wasted by the parameter based admission control solutions. To overcome this problem, different measurement based admission control methods were proposed. These methods still require a traffic descriptor for the new flows which is communicated to each node along the route at connection establishment time. (Per-hop measurement and admission decision is performed.) The nodes however use the traffic descriptor only to determine the resource requirement of the new connection, while the resource requirement of all existing connections is determined by measuring certain parameters of the live traffic. Two aspects shall be considered to characterise a per-hop measurement based admission control method. The traffic descriptor that is used to characterise a new flow, and the measurement and decision methodology that is applied. As pointed out in [32] and [33], components belonging to these two aspects can almost arbitrarily be combined to define a particular admission control scheme.

There are methods requiring only the peak rate of new connections [34, 33, 35], methods based on token bucket characterisation of sources [36, 32] while [37] proposes a method which does not need any parameter from the source.

With regards to the traffic measurement and load estimation process, some methods are based on firm mathematical foundations [34] while others are more based on heuristics [36]. Ongoing traffic can be measured per-flow [37], per traffic aggregate [33], but [35] shows that significant bandwidth savings can also be achieved if ongoing traffic is measured based on intelligently formed flow groups.

Surprisingly, given the wide variety of proposed methods they are capable of achieving very similar performance [33].

Some researchers argued that appropriate QoS can be provided with much lower implementation cost if the measurements are performed end–to–end, and not at each particular node along the path the connection takes [7, 30]. These approaches are based on active end–to–end measurement, and they require the sources to inject a probe stream into the network, and measure the loss rate experienced by this stream. The admission decision is based on the measured loss rate.

The method I propose and evaluate in this dissertation is also based on end–to–end measurements but differs from the above class in the measurement methodology. It does not rely on sending probe streams, rather it relies on *passive* end–to–end measurements that is it measures the performance of ongoing connections between two particular endpoints. In addition, not just loss, but also delay measurements can be incorporated into the decision process.

Figure 3.2 depicts the admission control solutions identified in this section.

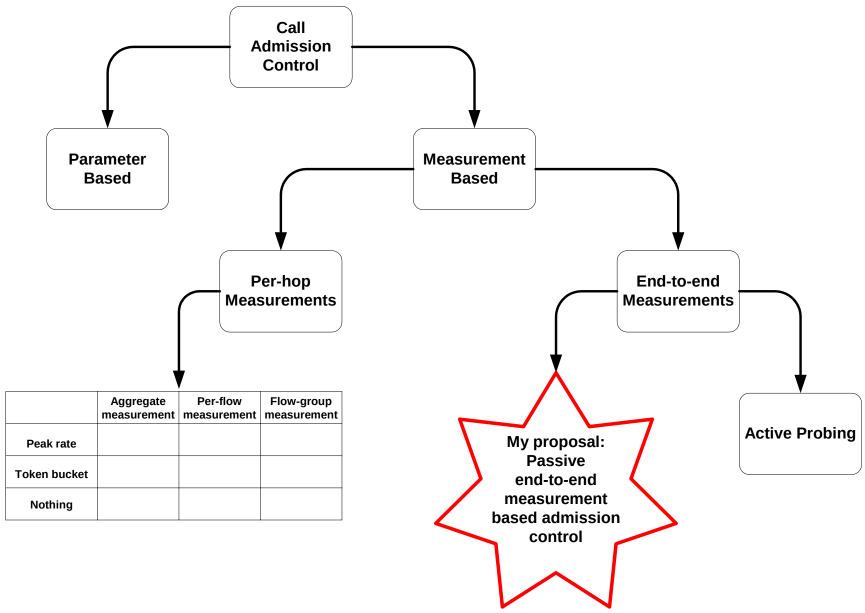

Figure 3.2: Taxonomy of call admission control methods

3.2 The Call Admission Control Method

This section together with the extension presented in 3.3.5 proposes a toolbox which consists of a basic method using a simple constant threshold–based decision process, and a couple of add–ons (statically and dynamically adaptive thresholds, actions in case of detecting lost measurement packets) which can be deployed in more demanding network scenarios. Based on the Quality of Service the network operator would like to provide and the network utilisation level it targets, the operator can decide which elements of this toolbox needs to be deployed and how those elements shall be configured.

In the network depicted in Figure 3.1, telephony calls are traveling through the core network between pairs of IP Telephony Gateways. There are hundreds of simultaneous calls between any two IP Telephony Gateways. The base of the new call admission control method is that the gateways collect aggregate statistics (e.g., packet loss ratio) about all the ongoing calls, and regularly share this information with the peer gateway. When a new call arrives at the gateway, it compares the latest available statistics with the target value set for the telephony calls, and accepts the call if the performance indicators representing the quality of the transmission path towards the target gateway are above the preset threshold. The target values can be set in a way that the tolerable delay is not exceeded by the packets of admitted voice sessions, and the packet loss suffered by the individual voice calls is below the limit, which can be compensated by the voice codec.

As far as the start–up phase is concerned, when no information about the core of the network is available, a reasonable assumption is that a core network can handle thousands of simultaneous calls, so the first few calls can for sure be admitted without compromising the performance.

This method of course can not provide zero loss for the admitted sessions. Zero loss however is not a requirement for the majority of the applications. For example state of the art voice coders can compensate relatively high frame error rates. The error concealment unit of Adaptive Multirate (AMR) speech codec can provide undistorted speech up to 1% frame error rate, and it can provide acceptable speech quality up to 10% frame error rate [38].

A summary of actions involved in the call admission control can be followed in Figure 3.2. The *receiving gateway* is delivered an IP packet encapsulating a voice frame. Before sending the voice frame to the destination access network, the gateway determines which gateway have sent this packet (based on the source IP address in the packet header), and which call the frame belongs to (Based on the call identifier included in the packet header). It checks the frame sequence number to calculate the number of packets that have been lost since the previous successfully received packet of the call, and updates the packet loss counter corresponding to the sending gateway. It also checks the timestamp to calculate the transmission delay. At the end of each measurement period the receiving gateway calculates the average packet delay measured during the period, and sends a control packet to each sending gateway to inform the senders about the loss and delay statistics of their calls. If the voice frames are carried in RTP frames, the control information can be conveyed by the control protocol RTCP. Note however that RTCP establishes a control stream for each individual voice call, so using RTCP means that as many control packets needs to be exchanged between the gateway pairs in each measurement interval, as many calls are active in that particular interval. In the current implementation, one control packet is sent reporting aggregate statistics about all the calls which are active between the gateway pairs. This approach very much limits the overhead resulting from the measurement reports. The format of the control packet is depicted in Figure 3.3.

After receiving a voice frame from the access network, the *s*ending gateway determines which call the packet belongs, and it increases the frame sequence number of that particular call. Finally the gateway encapsulates the voice frame into an IP packet. The encapsulation can for example be

Header
Measured Loss
Measured Delay

Figure 3.3: Format of the Control Packets

Sending side

Actions per voice packet

```
1. Determine callID
2. Seqno[callID]++
3. Determine timestamp
4. Build frame
5. Send frame
```

Actions per control packet

```
1. controlpacket.delay=sumdelay/packetcount
2. sumpacket=losscount+packetcount
3. controlpacket.loss=sumpacket/packetcount
4. Send control packet
```

Receiving side

Actions per voice packet

```
1. Determine Sending Gateway
2. Determine CallID
3. losscount=losscount+(seqno[callID]-packet.seqno)
4. packetcount++
5. delay=now-packet.timestamp
6. sumdelay=sumdelay+delay
7. Forward frame
```

Actions per control packet

```
1. lossrate=EWMA(lossrate,controlpacket.loss)
2. actdelay=controlpacket.delay
```

Figure 3.4: Pseudocode of the Call Admission Control Algorithm

done using RTP [39] framing. The only requirement from the framing protocol is to provide sequence numbering, and to allow the identification of individual calls. The destination address field of the IP header is set to the address of the receiving gateway. The receiving gateway can for example be looked up in a static routing table.

Upon receiving a control packet from the remote gateway, the state variables **"lossrate"** and **"actdelay"**, which represents the loss and delay suffered by the voice packets of the sending gateway is updated, typically by calculating a weighted average of the latest measurement result, and the previous value.

The actions taking place when an end user initiates a new voice call can be followed in Figure 3.5. Note that the message names appearing in the figure don't imply the use of any particular call control protocol. The end–to–end measurement based admission control method can work together with all existing call control signalling solutions.

A connection setup message travels through the access network of the calling party, and arrives at the first IP telephony gateway. The gateway derives the address of the remote gateway from the address of the called party received in the connection setup message. It compares the actual value of loss and delay measurements towards the remote gateway with the preset thresholds (losstarget & delaytarget). If the actual values are below the thresholds, the new call is accepted, and the connection establishment message is forwarded to the remote gateway (Step 3 of Figure 3.5.a). If any of the current figures is above the respective threshold, the new call is rejected (Step 3 of Figure 3.5.b). When making the call admission control decision

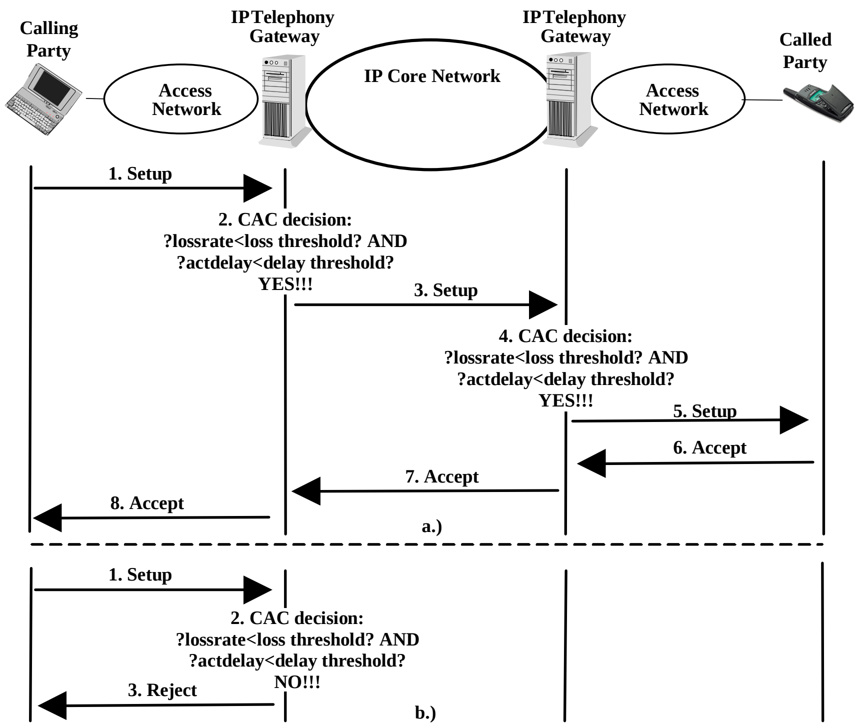

Figure 3.5: Sequence of actions for successful (a.) and unsuccessful (b.) connection establishment

(Step 4), the second gateway uses the measurement reports submitted by the first gateway about the ongoing traffic it receives from the second gateway.

As it is clear from the presentation above, the computational overhead of the method is very small. The method puts burden only on the edge routers (IP telephony gateways). A counter needs to be maintained for each active flow to be able to determine the packet losses. Three more counters are needed, one to calculate the number of packets arriving in one measurement interval, a second one to accumulate the number of packet losses, and a third one to accumulate the delay values.

The applicability of the proposed method is not constrained to IP Telephony gateways. IP will also be used as a transport technology in the core network of 3G mobile systems. There will be an IP tunnel established between the mobile network nodes (SGSN and GGSN) over the IP network upon arrival of a new packet data session. The tunnel establishment will be controlled by a mobile network specific protocol. The proposed end–to–end measurement based call admission method can easily be implemented as an addition to this mobile network specific tunnel management protocol, and can therefore be used to avoid overloading the IP network.

3.2.1 Enhancing the CAC Method using Adaptive Thresholds

If the network is used mainly for voice traffic, or there is a virtual private network type of arrangement for interconnecting the voice gateways, one can exploit the fact that it is very easy to characterise the daily profile of the phone traffic. Periods of modest network load are intercepted with periods of heavy traffic during the so–called busy hours. The gateways can be configured based on the daily traffic profile. A lower CAC threshold can be configured for the busy hours.

A more general solution would be to continuously change the CAC threshold. Upon receipt of a new loss report the sending gateway compares the value to a second threshold. If the latest loss report is above the second threshold, the gateway selects a smaller CAC threshold. It uses this stricter CAC threshold until the average loss reported by the remote gateway decreases to the desired value.

These enhancements can be beneficial in case of varying amount of offered traffic.

3.2.2 The Stationarity of Internet Path Properties

The method presented here uses on–line measurements to drive the admission control that is it tries to predict future network behaviour based on measuring the past. This approach can be successful in large scale networks if network measurements are stationary. [40, 41] analyses a large and diverse set of measurement data to address this issue in the context of the public Internet. They argue that three types of "constancy" need to be distinguished: mathematical, operational and predictive.

A dataset of network measurements is *mathematically steady* if it can be described with a single time–invariant mathematical model. However, many mathematical non–constancies are in reality irrelevant to protocols. For instance, if the loss rate on a path was completely constant at 10% for thirty minutes, but then changed abruptly to 10.1% for the next thirty minutes, one would have to conclude that the loss dataset was not mathematically steady, since its fundamental parameter has changed; yet one would be hard–pressed to find an application that would care about such a change. The authors of [41] call a dataset *operationally steady* if the quantities of interest remain within bounds considered operationally equivalent and a dataset is called *predictively steady* if past measurements allow one to reasonably predict future characteristics.

As far as the measurement data is concerned, it was collected in two periods during winter of 1999 and 2000, from a measurement network comprising 31 and 49 hosts, around 80% of them being located in the USA and half of them being university or research institute sites. The measurements utilised public traceroute servers [42] and the NIMI measurement infrastructure [43].

Loss Constancy

The packet loss over the two whole data sets were quite low, 0.87% and 0.60%. However, traces experienced a wide range of loss from no loss at all to loss ratio exceeding 10%. It was found that loss rates in the two directions on a path are only weakly correlated, with a coefficient of correlation of 0.10 for the 70% of traces that suffered some loss in both directions. However, the logarithms of the loss rates are strongly correlated (0.53), indicating that the order of magnitude of the loss rate is indeed fairly symmetric. The investigation found that it is better not to consider the loss process but the *loss episode* process i.e., the time series indicating when a series of consecutive packets (possibly only of length one) were lost. The loss episode interarrivals for each trace are consistent with exponential distributions, even though the mean loss episode rate in the traces varies from 0.8%–2.7%, and this in turn argues strongly for Poisson loss episode arrivals. It was also confirmed that loss run lengths in packets often are well approximated by geometric distributions.

Mathematical constancy was investigated by determining the length of regions in the traces during which the loss episode rate can be regarded as steady. It was found that more then half of the traces were steady during the whole 1 hour long measurement period, but the length of steady period is much smaller if the analysis is restricted to those traces with loss episode rate of at least 1%. The average number of change free regions was 5 over all traces and it was 20 over the traces with more then 1% loss. In order to investigate operational constancy, the loss rates were partitioned into the following categories: 0–0.5%, 0.5–2%, 2–5%, 5–10%, 10–20%, and 20+%. These categories try to capture the qualitative notions such as “no loss”, “minor loss”, “tolerable loss”, “serious loss”, “very serious loss”, and “unacceptable loss”. If we only care about constancy of loss viewed over 1 minute periods, then about two–thirds (57–71%) of the time, we find we are in a constancy period of at least an hour in duration.

Finally, the question is to what degree can a predictor predict the length of the next loss–free run. Three different types of predictors were tested: moving average, exponentially–weighted moving average, and the S–shaped moving average estimator [44]. This last one is a class of weighted moving average estimators that give higher weights to more recent samples. For each of these estimators there is a parameter that governs the amount of memory of past events used by the estimator. The main conclusion is that virtually all of the estimators perform about the same — the parameters do not matter, nor does the averaging scheme.

Delay Constancy

The study of delay constancy was confined to round–trip time (RTT) measurements. The first observation is the presence of delay spikes which are intervals (often quite short) of highly elevated RTTs. There is a separate analysis presented in [41] for the delay spikes and for the body of the delay distribution. Analysis of the body of the delay distribution reveals that overall, delay is less steady than loss, and that, while there is a wide range in the length of steady delay regions, in general delay appears well described as steady on time scales of 10–30 minutes. The RTT spike episodes are quite short–lived, the median duration was 150ms. Further on, it was found that the RTT spike episodes are even more steady than loss episodes: the process is steady across the entire hour 75% of the time. In addition, the interarrivals between spikes are well–modeled as independent, identically distributed exponential. Operational constancy is assessed using the categories defined by ITU–T in [45]. It was found that more than half of the traces have maximum change free regions under 10 min, and 80% are under 20 min so not only are packet delays not mathematically steady, they also are not operationally steady.

The same three types of estimators tested for loss prediction were also tested for RTT prediction. The estimators again all performed virtually identically, and (in contrast to the loss prediction) their performance is very good.

Routing Constancy

Two aspects of routing constancy were investigated: *prevalence* and *persistence*. Prevalence means how often the most commonly occurring ("dominant") route is observed, while persistence reflects the number of consecutive traceroute measurements each observed the same route. For the NIMI routes, 78% always exhibited the same path, and 86% of the routes had a prevalence of 90% or higher. For the public servers, the corresponding figures are 73% and 85%, respectively. Even for the 15% of routes for which the dominant path does not completely dominate, it still is almost always observed the majority of the time. Most paths are persistent over time scales of many hours to days, but a fair number of paths are persistent only for shorter time scales. Evidence was found that a total of about 1/3 of Internet routes in general, and 1/6 of the NIMI routes, are short–lived.

Consequences on the Design of End–to–end Measurement Based CAC

The answer to the question "how steady is the Internet?" depends greatly on the particular aspect of constancy and the dataset under consideration. However, it appears that for all three aspects (loss, delay, routing) which are relevant to end–to–end measurement based call admission control one can generally count on constancy. This observation is definitely encouraging when the question is the applicability of measurement based control techniques over the Internet.

In particular, Internet routes appear to be very stable. This is an important prerequisite to the success of the end–to–end measurement based call admission control. Given that the approach is core–stateless and the gateways "blindly" measure loss and delay over the path interconnecting them without being able to detect changes in the routes, the stability of routing helps assuring that the actual measurement data is characterising the path which will later be used by the admitted traffic.

Another general finding is that almost all of the different classes of predictors frequently used in networking (moving average, EWMA, S-shaped moving average) produce very similar error levels. This fact simplifies the design, and more importantly, the testing of measurement based control solutions because paying much attention to the peculiarities of predictors becomes unnecessary.

The fact that the traces can be divided into more or less change free regions where a particular property can be regarded as steady argues in favor of applying adaptive thresholds as suggested in Section 3.2.1.

3.3 Performance Evaluation

This section presents detailed numerical results about the performance of the proposed call admission control method in different networking scenarios. The call admission control method is implemented in ns-2 [46]. The method is tested assuming a core network, where support for differentiated services is available, and the applicability of the technique in a best effort network is also considered. In the latter case, not only the voice communication quality but also the effect of the admission controlled traffic on legacy TCP flows is investigated. Table 3.1 provides a summary of the different cases that were simulated in order to characterise the proposed method.

3.3.1 Can the evaluation be done using analytical models?

There are 3 broad classes of equally useful research methodologies (measurement, simulation and analytical modelling) when it comes to investigating telecommunication systems. Analytical models were perfect for describing legacy circuit switched telephone networks, but the complexity of the network and the number of traffic types carried in today's IP networks makes their proper analytical description a very difficult (if not impossible) task.

For per-hop measurement based solutions there are a couple of approaches based an quite elaborate mathematical models. See for example

Table 3.1: A Summary of the Simulation Scenarios

Diffserv Network		
Case	**Controlled Parameters**	**Page**
verifying the configuration of the CAC	measurement interval	38
simple two–node network, single gateway pair	On/Off or CBR source, link load, link speed, buffer size	41
simple two–node network, two gateway pairs	CAC threshold, call arrival rate	45
simple two–node network, varying offered traffic	a second gateway pair joins in, Pareto On/Off sources join in, the call arrival rate changes	46
Multihop topology	varying number of bottlenecks	48
Adaptive CAC threshold	busy–hour based adaptation, CAC threshold based adaptation	49
Best-effort Network		
Case	**Controlled Parameters**	**Page**
motivating example	TCP traffic mixed with unregulated voice traffic	52
simple two–node topology, CAC uses only loss measurements	# of parallel TCP sessions, link speed	54
simple two–node topology, CAC uses only loss measurements	CAC threshold, buffer management	55
CAC uses both loss and delay measurements	measurement interval, multihop topology	58
loss on the feedback path	feedback packet loss rate, loss compensation method	63

[34]. Grossglauser and Tse [47] presents an analytical framework for studying the performance of measurement based admission control schemes under measurement uncertainty and flow dynamics. On the other hand, there are a handful of simulation studies as well. One particular work worth noting is [33], which is a comparative simulation study of different per-hop measurement based admission solutions, with an interesting conclusion saying that the performance of methods based on complex mathematical foundations do not much differ from the performance of admission control methods based on the simplest possible heuristics.

In the field of end-to-end measurement based solutions, the literature thus far covers only different flavours of the active, probe packet based method. Three analytical models are reported. [30] presents an analytical model for both CBR and VBR sources but simulations reveal that the model is reasonably accurate only for CBR sources. [48] extends an analytical framework previously proposed for per-hop measurement based admission control to end-to-end measurement based admission control. The paper discusses only delay provisioning and the proposal has some inherent deployment problems. The solution requires either synchronised clocks at the decision making endpoints or it requires the routers in the network core to update a field in each packet header with the amount of delay the packet suffered at that particular hop. An analytic model is presented for the Explicit Congestion Notification based version of the active end-to-end measurement based admission control in [49].

These analytical models are applicable only to specific approaches to the active end-to-end measurement based admission control, and the models give aggregate statistics only, they are unable to answer questions about the performance of admitted individual calls. On the other hand, the most cited investigation, which is a comprehensive and comparative study of many different approaches for probe based admission control is based solely on ns-2 simulations [50].

Based on this literature survey, we can conclude that simulation is a more promising approach when aiming at a comprehensive simulation study of the proposed admission control solution, and that simulation based approaches are acceptable for the international research community.

3.3.2 Simulation Model

Since the method presented here is targeted primarily for supporting voice users, an important aspect of the simulation is the voice traffic model. Two, widely accepted, voice traffic models are used. Constant Bit Rate (CBR) voice sources generate 70 bytes long packets in each 20 ms. This packet size corresponds to the standard, uncompressed RTP/UDP/IP framing of voice data. The On/Off voice sources also generate 70 bytes long packets, but only in the ON periods, while in the OFF periods no packet is sent. The duration of ON and OFF periods are exponentially distributed, the mean length of ON periods is 352 ms, and the mean length of OFF periods is 650 ms, as suggested in [51]. The voice call arrival process is Poisson and the call holding time is modeled by an exponential distribution with mean of 90s.

The TCP flows use the Reno version of the congestion control algorithm, which is widely deployed. TCPs are fed by an FTP application simulating the download of a large file, which means that TCP has data to send during its entire lifetime. TCP generates 1000 bytes long packets. Each TCP source and sink is connected to the first hop router via a dedicated 100 Mb/s link. The propagation delay of the links between the sources and the first hop router is set to 2ms. On the other hand, the delay of the link between the first hop router and the sinks is set to a different value for each connection. This ensures that the TCP flows have different round trip time.

The literature uses the so–called loss–load curves to demonstrate the performance of a particular measurement based admission control technique. This curve shows the average loss suffered by the flows subjected to the admission control. Such an average value is quite meaningless when it comes to packet switched compressed voice communication. A state of the art voice codec can completely conceal quite high loss rates (typically up to 1%), so for voice communication the relevant question is that given a CAC method, how many percent of the individual voice flows exceed a certain loss threshold. In this document the figures show the fraction of individual voice flows which suffered more then 1%, more then 3%, and more then 10% loss. With a typical voice codec, the 10% value can be regarded as the fraction of the calls that provided unacceptable voice quality, while the 3% figure approximates the fraction of voice sessions, which suffered noticeable performance degradation.

3.3.3 End-to-end Measurement Based Call Admission Control in a Diffserv Capable IP Network

Configuration

If the core network supports service differentiation for traffic aggregates based on the differentiated services concept, the low delay and jitter for voice traffic can be guaranteed by marking voice packets to receive preferential treatment [52]. In this case, the required low delay for voice packets can be provided by configuring the routers to assign small buffers for the traffic class, where the voice calls are carried, which results in low end–to–end delay. In this configuration the gateways collect only packet loss statistics.

Before deploying the method proposed here one needs to determine the length of the measurement interval, or with other words the frequency of sending control packets. Longer measurement interval means more stable results, however it also means slower reaction to changes in network load. The control packets basically represent the overload associated with the method, and it should be kept as low as possible, which motivates the use of longer measurement period.

Let us assume that RTCP sessions are used to carry the measurement reports on a per call basis. A typical voice codec with 20ms frame length generates about 50 packets in a second. The RTP specification says that the bandwidth consumed by the RTCP packets is not allowed to exceed 5% of the session bandwidth. The above two constraints would limit the feedback frequency to a maximum of two packets per second, that is a minimum measurement interval of 0.5 seconds. Calculating with the typical voice packet size of 70 bytes, a 15 Mb/s link can carry a maximum of ~26700 packets in a second, so even the 1 second long measurement interval is a little bit short for reliably measuring packet loss ratio of 0.01–0.001.

As mentioned earlier, an important advantage of the scheme proposed here compared to previous end–to–end measurement based CAC methods is that it measures live network traffic continuously instead of just measuring a sequence of probe packets dedicated to a specific call. Therefore it can take into account a longer history of measurements without sacrificing the timeliness of its response. As discussed in Section 3.2.2, all of the different classes of predictors frequently used in networking (moving average, EWMA, S–shaped moving average) produce very similar error levels therefore in the prototype implementation in the simulator I picked EWMA, (The sending gateway calculates the new loss result as a weighted average of the previous value and the information received in the latest control packet.) and I have not tested with other type of predictors.

Based on the above mentioned consideration of low overhead, typical simulated link sizes, timely reaction to incipient congestion, and stable measurements, the choice is a measurement interval of 1 second and equal (0.5) weighting of the latest report and the previous value when calculating the new loss estimate. Simulation results are presented to justify this decision.

Figure 3.6 shows the dependence of link utilization on the measurement interval, and Figure 3.7 depicts the fraction of calls that suffered more than 1% packet loss. The utilisation values are calculated for the final 1000 seconds of the simulations. Four sets of simulation results are presented. All four are run with a single link connecting one gateway pair. The link speed is set to 1 Mb/s or 15 Mb/s, the CBR voice model is used in one experiment, and the On/Off model is configured for the other 3 simulations. The call acceptance threshold is set to 0.01 when running with the slower link, and it is set to 0.005 for the 15 Mb/s link. The average flow inter–arrival time is 1.2 seconds for the slow link case and it is 0.09 second for the 15 Mb/s link. The corresponding blocking probabilities are summarised in Table 3.2. In the fourth experiment (on_off15Mb_par) the CAC method is

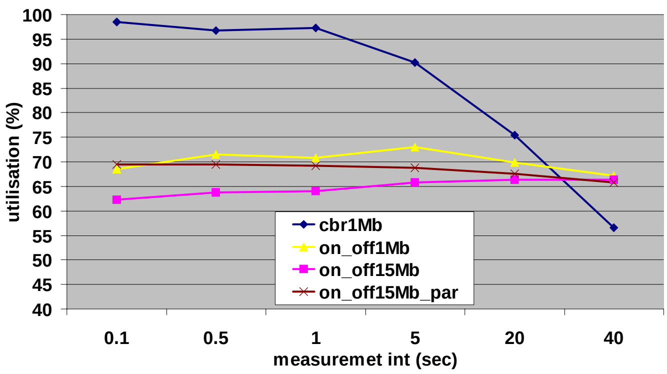

Figure 3.6: Effect of varying the measurement interval on the utilisation

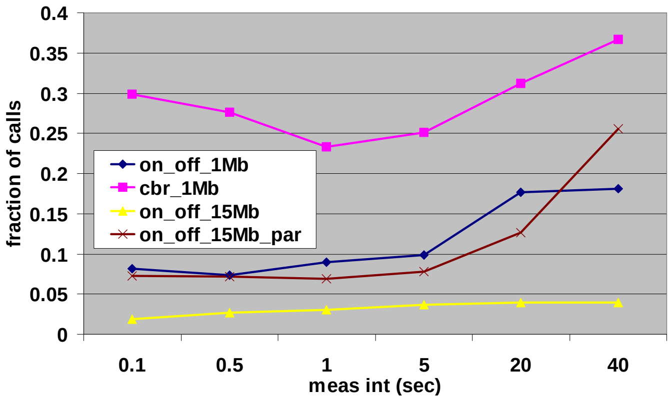

Figure 3.7: Effect of varying the measurement interval on the performance of voice calls

disturbed by 150 Pareto On/Off sources[1]. The background sources start at the beginning of the simulations.

Measurement intervals in the range of 0.1 and 1 second yield very similar performance. Above that range the performance of the method degrades. The explanation is simple: while a measurement is valid there can be too many calls admitted if the measurement interval is long which causes the degradation of the per–call performance. As a conclusion, the choice of 1 second for the measurement interval is reasonable but the expected range of call inter–arrival time needs to be taken into account when making this decision.

[1]The Pareto distribution has a shape parameter of 1.2. The source generates 125 bytes long packets at a rate of 64 kbyte/s. The average ON and OFF time are both set to 500 ms.

Table 3.2: The average blocking probability measured over the four traffic cases

Measurement interval (s)					
0.1	**0.5**	**1**	**5**	**20**	**40**
0.2578	0.2552	0.2505	0.2607	0.2949	0.3042

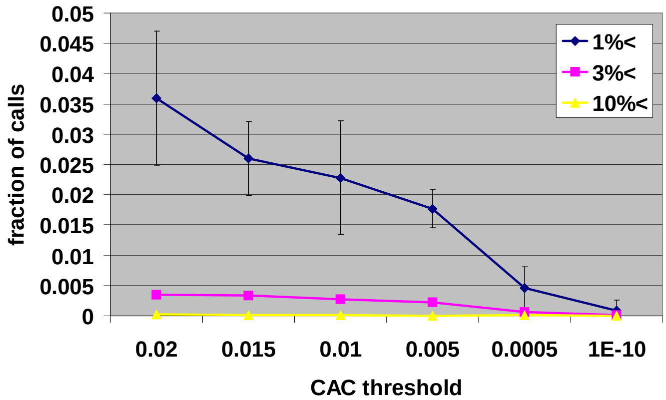

Figure 3.8: Performance of On/Off calls

One Pair of Gateways Connected by a Single Link

This simulation setup contains 2 nodes, which are interconnected by a 2 Mb/s link. The first node initiates the voice calls, and the second one gathers the packet loss statistics and reports it back. To assess how well the aggregate loss used in the call admission control decision translates into per–flow performance figures, per–flow statistics about the packet loss suffered by each individual call are collected. The average network utilization for the last 1000 seconds of the simulations is also measured.

Figure 3.8–3.12 depict the ratio of individual calls, which suffered more then 1%, 3% and 10% packet loss. Six sets of simulations were run for both the ON/OFF and CBR voice model with the CAC threshold ranging between 0.02 and 10^{-10}, and the figures demonstrate that the proposed method is capable of ensuring the desired low packet loss ratio for the individual voice calls. The number of voice calls suffering more then 1% packet loss ratio was below 1% with all 6 thresholds for the CBR traffic (Figure 3.9). The behaviour of ON/OFF sources is also acceptable, since the percentage of flows hit by more then 1% packet loss is very close to 0 using the decision threshold of 0.0005 (Figure 3.8). A nice property of the ON/OFF sources is that basically none of them exceeded the 10% packet loss ratio. (Recall that 10% packet loss ratio can be regarded as an upper bound for producing acceptable voice quality.)

These simulations were run with modest offered traffic, the average flow

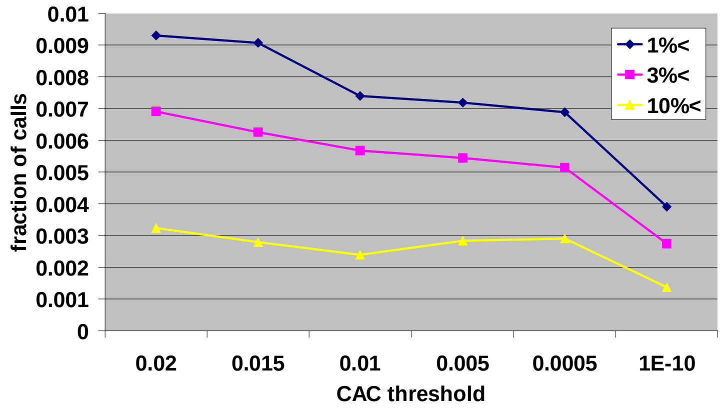

Figure 3.9: Performance of CBR calls

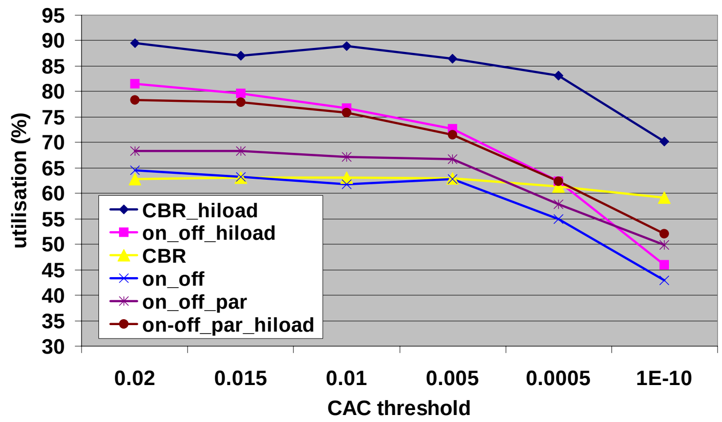

Figure 3.10: Average link utilization

inter–arrival time was 0.68 seconds in the ON/OFF case, and 1.9 seconds for the CBR case. The call blocking probability was 0.9% with the CBR sources, and 1.2% with the ON/OFF sources when the CAC threshold was set to 0.01. The corresponding utilization curves are plotted in Figure 3.10.

The method is capable of filling the link up to 65%. If the CAC threshold gets smaller, the utilization degrades more quickly with ON/OFF sources. Since the link speed is relatively low, a short peak of sources being simultaneously ON can drive the measurement result above the threshold, and it stays there for a relatively long time. Network operators might have different concerns, one might have cheap bandwidth, others might want to offer a high quality communication facility. By varying a single parameter (CAC threshold), the method can be tuned to provide the preferred trade–off between link utilisation and voice quality.

The next set of simulations test the method under heavy load. The average flow inter–arrival time is 1 second for the CBR sources, and 0.4

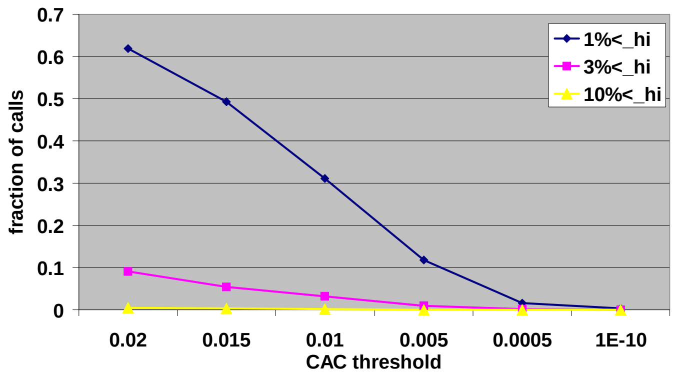

Figure 3.11: Performance of On/Off calls (high load)

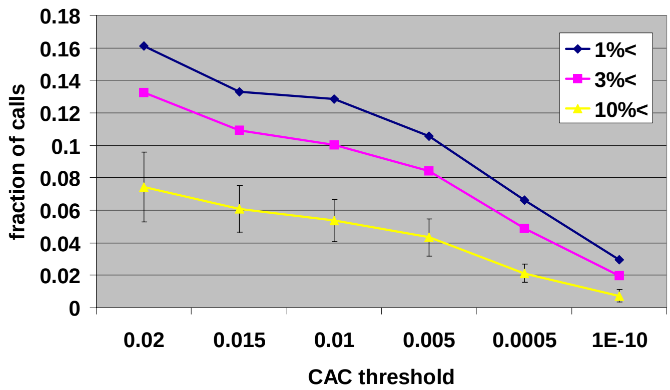

Figure 3.12: Performance of CBR calls (high load)

second for the ON/OFF model. Figure 3.11 and 3.12 depicts the per–flow loss characteristics, while Figure 3.10 shows the link utilization.

The call blocking probability is 28% with the CBR model and 29% with the ON/OFF sources when the CAC threshold was set to 0.01. Comparing the results with the modest load case, we can observe that much lower CAC threshold value is needed to provide really good quality for the voice calls. This dependence of the optimal threshold from the offered load is impractical, but even with the CAC threshold of 0.0005, the average link utilization is above 80% for CBR traffic, and around 60% for ON/OFF sources. Note once again that the 10% loss bound is only exceeded by a very small fraction of the calls in all settings.

The next set of simulation is run with On/Off voice model, and 20 Pareto On/Off sources as background traffic. In the first experiment, the average call inter–arrival time is 1.25 second, in the second experiment it is 0.8 second. Maximum 1 voice call suffered more then 10% loss in all the simulation runs. The 1% and 3% loss values are depicted in Figure 3.13. The utilisation

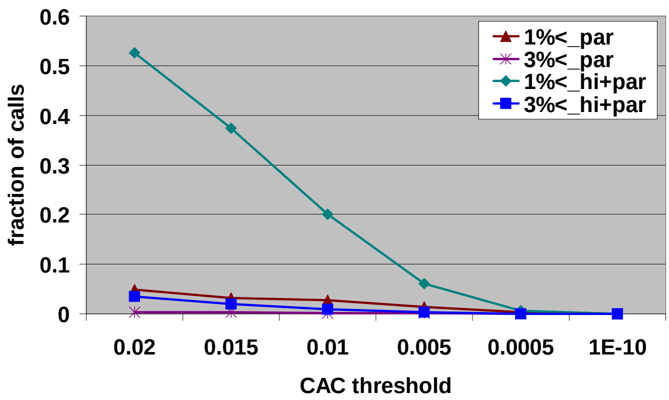

Figure 3.13: Performance of On/Off calls with Pareto background traffic

Table 3.3: The effect of buffer size

Buffer	**Utilisation**	**Block**	**1%<**	**3%<**	**10%<**
6	1533508	0.2939	0.3108	0.0326	0.00234
9	1722958	0.2142	0.233	0.0223	0.0016
12	1762219	0.1877	0.1861	0.0153	0.001
24	1801430	0.1719	0.1749	0.0158	0.0009

curve can be found in Figure 3.10 (on_off_par & on_off_par_hiload). The call blocking probability is 20% with the higher offered traffic and 3% with the lower call intensity when the CAC threshold was set to 0.01.

So far the call admission control method was tested with very limited buffer sizes. Clearly, there is a trade–off between delay and packet loss ratio/link utilization. Table 3.3 summarizes the simulation results for the high load case if longer queues are tolerated. A 24 packets long queue represents a worst case per–hop delay of 6.72 ms on the 2 Mb/s link. The CAC threshold is set to 0.01. The call inter–arrival time is 0.4s for the On/Off calls, and it is 1s for the CBR flows.

Increasing the buffer capacity results in more and more calls being admitted into the network, and those admitted calls experience a slightly better loss performance as well. However, it is clear that decreasing the CAC threshold has much more positive effect on the per–call loss figures.

The 2 Mb/s link capacity, which has been used in the experiments above does not really represents the link capacity available in networks today. It was chosen to obtain reasonable simulation time, but it is believed to be sufficient to assess the behavior of the proposed technique. Table 3.4 summarizes the simulation results for higher link speeds. The CAC threshold is set to 0.005. The load is expressed as the average flow inter–arrival time. Higher link speed leads to smaller blocking probabilities, and similar communication quality.

Table 3.4: The effect of multiplexing

Link	**Load**	**Block**	**1%<**	**3%<**	**10%<**
2 Mb/s	0.7875s	0.0099	0.0059	0.0008	8.83E-05
15 Mb/s	0.105s	0.0006	0.0054	0.0007	7.01E-05
45 Mb/s	0.035s	0	0.0076	0.0009	4.60E-05

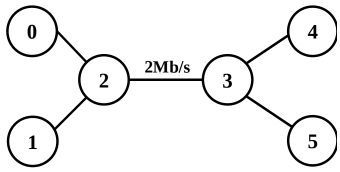

Figure 3.14: Network Configuration with Two Gateway Pairs

Multiple Gateway Pairs Sharing a Bottleneck Link

The goal of the simulation presented in this section is to study the interaction of two IP telephony gateways if both run the proposed call admission control method. The simulation scenario is depicted in Figure 3.14.

The bandwidth of the links is set to 3 Mb/s, except of the link between node 2 & 3, where the capacity is 2 Mb/s. This ensures that the link between node 2 and 3 is the bottleneck link, and this link is shared by the two gateway pairs. Node 0 generates voice calls towards node 5, while node 1 communicates with node 4. Node 4 & 5 independently collect aggregate statistics about the packet loss ratio of their own calls during 1 second intervals, and the smoothed measurements results (EWMA weight is set to 0.5) are used for the call admission control decision in node 0 and node 1.

The goal of the first experiment is to study the case when the two gateway pairs use different threshold for the CAC decision. ON/OFF sources are used in the experiment, and the average flow inter–arrival time was 1.35 seconds in both node 0 and node 1. Table 3.5 presents the simulation results.

Table 3.5: Simulation results for the multiple gateway case with asymmetric CAC thresholds

Node 0					
Case	**Threshold**	**Blocking**	**1%< loss**	**3%< loss**	**10%< loss**
1	0.01	0.0067	0.0143	0.0016	0.0003
2	0.01	0.0061	0.0122	0.0012	0
3	0.01	0.0002	0.0011	0.0002	0
Node 1					
1	0.01	0.0098	0.0156	0.0012	0.0002
2	0.005	0.0406	0.0124	0.0017	0
3	10E-10	0.4158	0.001	0	0

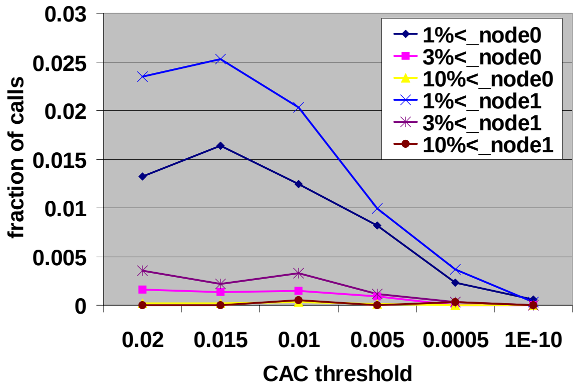

Figure 3.15: Per–call performance (two gateway pair scenario)

If the two gateway pairs use the same CAC threshold, they experience the same performance, and each gateway receives its fair share of the bandwidth of the bottleneck link. (The same offered load results in the same blocking probability.) In the asymmetric cases, the gateway with the higher call acceptance threshold monopolizes the link, and gets almost all of its calls through. This is indicated by the huge difference between the blocking probabilities seen by node 0 and node 1 in case 3. Looking at the loss figures, it is clear that the bandwidth sacrificed by node 1 by applying stricter CAC threshold does not pay off. The per–call performance of the two gateways is indistinguishable. The loss figures in case 3 are better compared to the case depicted in Figure 3.8 when a single gateway offered the same amount of load. This is simply because the number of calls admitted to the link is smaller in the two gateway case due to the very high blocking rate of node 1.

In the next experiment, both gateways use the same CAC threshold, but they initiate different amount of calls. The average call inter–arrival time is 0.95 second for node 0, and it is 2.45 second for node 1. Figure 3.15 depicts the loss statistics of the calls. We can observe that the two gateways do not disturb each other. Their calls experience very similar loss characteristics over all the CAC threshold values. The utilization of the bottleneck link is around 65%, the same as it is when only a single IP telephony gateway pair loads the network.

Experiments with Dynamically Changing Load

The goal of this section is to characterise the behaviour of the system when the offered traffic varies during the experiment. In the first experiment, we use the network topology of Figure 3.14. The traffic we measure flows between node 1 & node 4. During two 800 seconds long time periods the other gateway pair also generates traffic. The call inter–arrival time of the

Table 3.6: Summary of the experiments with dynamically changing network load

Experiment 1				
	Blocking	**1%< loss**	**3%< loss**	**10%< loss**
Normal load	0.0318	0.0383	0.003	0
High load	0.2608	0.2934	0.0285	0.0012
Experiment 2				
Normal load	0.0183	0.0305	0.0028	0.0009
High load	0.2664	0.3	0.0225	0.0003
Experiment 3				
Normal load	0.0239	0.0374	0.00275	0
High load	0.2996	0.2948	0.0273	0.0017

first gateway pair is 0.68 seconds, while the inter–arrival time for the background traffic is 1.1 second. In the second and third experiment, we use the single link network topology. In the second case 20 Pareto On/Off sources generate background traffic for the 800 seconds long periods, while in the third experiment, the call inter–arrival time of the IP telephony gateway decreases to 0.4 for two times during the simulation. The CAC threshold is set to 0.01 in all simulations.

Table 3.6 summarises the results. It shows the blocking probability for normal load and high load periods, and the fraction of individual calls, which exceeded the loss bound 1%, 3% and 10%. It is clear from the data that the CAC method reacts to the increased level of congestion as the increase of the blocking probability indicates. It is also clear however that the reaction is not aggressive enough since the per–call performance degrades. We can observe that very few (a maximum of 2–3) calls are rendered to unacceptable voice quality even in high load periods. To better understand the behaviour of the CAC method, the utilisation of the link is measured in subsequent 500 ms intervals between 800 and 4200 seconds of simulation time. Figure 3.16 depicts the utilisation values for the second experiment, but the graph was very similar for the other experiments as well.

It can be seen that when the offered traffic increases the CAC method is not capable of preventing a significant increase in the link load. To find out the reason for this, detailed per–flow statistics were collected about voice calls active in the normal load and in the high load period respectively. As we can see in Table 3.7, the CAC method keeps the loss ratio averaged over all the flows below the target value (CAC threshold=0.01) even in the high load case. However, due to the per–flow stateless scheduling in the network, the distribution of packet loss events is very unequal between the sources. At least 55% of the calls suffers no loss with normal load, and despite the low average figure, the maximum per–flow loss in the high load case is 36%.

Now we have average figures so we can compare the results produced

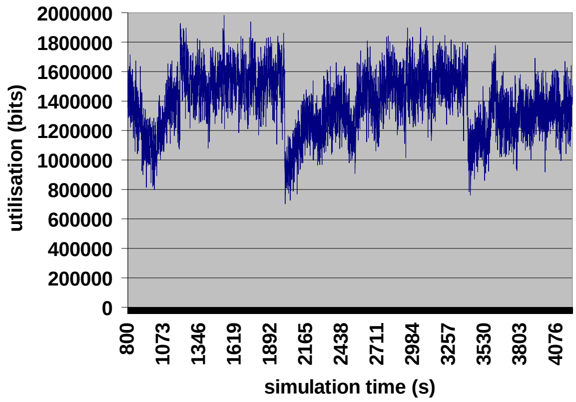

Figure 3.16: Link utilisation with varying offered traffic

Table 3.7: Detailed per–flow loss statistics in normal and high load situations

	Average	**Minimum**	**Maximum**	**55 perc.**	**90 perc.**
Normal load	0.0017	0	0.0726	0	0.0047
High load	0.0092	0	0.3684	0.0076	0.019

by this admission control method with results available in the literature. [33] compares the performance of traditional measurement based admission control algorithms. Taking a look at the loss load curves presented there we find that the end–to–end measurement based method is a little bit more conservative. The per–hop measurement based methods are capable of providing 0.001 average packet loss rate with around 90% link utilisation, while the method tested here can go up to 65%. The simulations presented in [33] were run with much larger buffers (160 packets), which makes the comparison a bit unfair. Such huge buffers are irrelevant to IP telephony because of the delay constraints. Note however the utilisation figures presented in Table 3.3 The figures demonstrate that the link utilisation achievable by the proposed call admission control method increases quite significantly if larger buffers, and consequently, larger delays are tolerated.

Testing in a Multiple Hops Topology

In the literature, CAC methods are usually tested in a single link topology, however, the characterisation of an end–to–end measurement based scheme can not be complete without considering topologies with multiple hops, and flows transferred through multiple bottleneck links. In this chapter we use the network topology depicted in Figure 3.17.

In the first experiment we have traffic between node0-node5, node1-node4, node9-node12 and node13-node14. In all pairs the first node acts as sending gateway, and the second one is the receiving gateway. The link

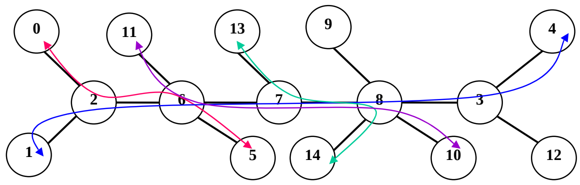

Figure 3.17: Network topology used for simulating with multiple bottleneck links

capacity is 2 Mb/s in the backbone, and 10 Mb/s in the other links. One exception is the link between Node6 and Node7, which is also set to 10 Mb/s, therefore it is not a bottleneck. Average flow inter–arrival time is 1.1 second for all four pairs. We would like to compare the performance of the calls travelling through one and three bottleneck links. In order to save space just condensed results are presented for two gateway pairs in Table 3.8. The average utilisation of the link between node2 and node6 was measured to be 69% during the final 1000 seconds of the simulation. As we can see, the CAC method negatively discriminates the gateway, which sends its traffic via a longer path. Comparing the result with those reported in [50], we note that the method proposed here exhibits approximately the same behaviour, which was described there in connection with an earlier approach to end–to–end measurement based admission control.

The setup for the second experiment is the same. The only difference is that the call inter–arrival time for the traffic between node13 and node14 is lower (0.9 second), and this traffic is not regulated by any call admission control procedure. Table 3.8 indicates that the unregulated traffic further deteriorates the performance of the long flow, but also demonstrates the advantages of admission control if we compare the first column of the first and fourth rows (0.06<<0.19). The experiment was repeated with running 20 Pareto background sources instead of the voice calls between node13 and node14, which yielded approximately the same performance degradation.

In the last experiment (Experiment 3. in Table 3.8), the first setting is used but the traffic between Node9-Node12 is removed, and it is replaced with the traffic flowing between Node11-Node10. In this experiment the link speed between Node6 and Node7 is decreased (2 Mb/s) to ensure that it is also a bottleneck for the new traffic flow. This new traffic flow travels through 2 bottleneck hops, so its performance confirms the correlation between the number of bottlenecks traversed by the flow and the performance degradation it suffers.

Experimenting with Adaptive CAC Thresholds

As presented in the previous sections the end–to–end measurement based approach is not robust enough. It can handle wide range of offered traffic

Table 3.8: Comparison between one–hop and three–hop calls

Experiment 1			
	1%< loss	**10%< loss**	**Blocking**
Node0-Node5	0.0619	0.0005	0.0476
Node1-Node4	0.1978	0.0009	0.1894
Experiment 2			
Node1-Node4	0.3078	0.0009	0.286
Node13-Node14	0.1908	0.0008	0
Experiment 3			
Node1-Node4	0.2925	0.3036	0.0015
Node11-Node10	0.1584	0.1497	0.0007
Node0-Node5	0.0344	0.0457	0.0003

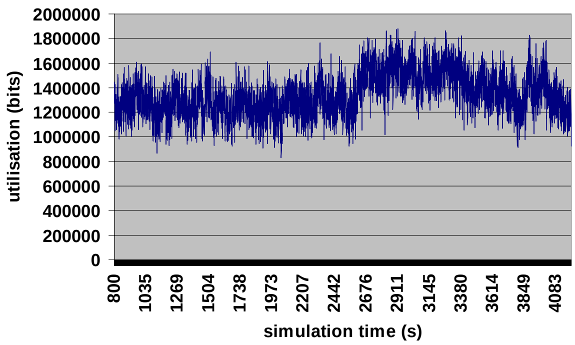

Figure 3.18: Effect of the busy–hour adaptation method on the link utilisation

but the optimal CAC threshold is different for high and low traffic load, especially if one is concerned with not only the communication quality but also the network utilisation. The CAC threshold should be adaptively set in a way that the increase in network load is prevented, and the measured loss rate is kept about an order of magnitude below the target loss value. This section presents simulation results using the adaptation methods described in Section 3.2.1.

The first simulation results demonstrate the benefits and limitations of the simple, busy–hour based solution. The simulation is run with the single link topology, and uses the settings of experiment 3 from Table 3.6. The only exception is that the gateway is configured to use a CAC threshold of 0.0005 during the first period of increased load.

Table 3.9 shows that the adaptation of the CAC threshold increases the blocking probability very significantly, and yields to very nice performance under both traffic load. It is also clear from the table (High load II) and from Figure 3.18 that this method is applicable only if the network operator can determine the periods of heavy network load in advance.

Table 3.9: The effect of the busy-hour based adaptive CAC threshold

	Blocking	**1%< loss**	**3%< loss**	**10%< loss**
Normal load	0.0233	0.0192	0.0031	0
High load I.	0.4441	0.014	0.0006	0
High load II.	0.27	0.267	0.0234	0.0014

Table 3.10: The effect of the second adaptation method

	Blocking	**1%<**	**3%<**	**10%<**
Normal load (w adaptation)	0.0679	0.0155	0.0021	0
Normal load (w/o adaptation)	0.0233	0.0192	0.0031	0
High load (w adaptation)	0.3179	0.0488	0.006	0.0009
High load (w/o adaptation)	0.2698	0.267	0.0234	0.0014

The method with continuous adaptation is tested in two simulations. Based on Table 3.7 we can devise a promising adaptation algorithm. In order to deliver high performance, the measured loss ratio shall be kept around one order of magnitude below the loss target. The first simulation uses the topology of Figure 3.14 and the setting of experiment 1 in Table 3.6. The exception is that both gateways implement the following adaptation: The CAC threshold is changed to 0.0005 upon measuring aggregate loss above 0.004, and switches back to 0.01 after measuring packet loss below 0.002. Table 3.10 summarises the simulation results. The adaptation of the CAC threshold leads to slightly higher blocking probability, and slightly lower utilisation for the normal load period, but the performance during high load periods improves significantly.

The last experiment uses the multihop topology of Figure 3.17. We measure the performance between Node1 and Node4. There is continuous voice traffic between node0 and node5. During 1200–2400 second of simulation time there are voice calls generated between node13 and node14, which are not subjected to CAC. Between 2000–3200 second of simulation time 20 Pareto On/Off sources generate traffic between node10 and node11. Finally the call inter–arrival time of the voice flows generated between node9 and node12 is decreased between 2800 and 3600 second. Table 3.11 compares the case when a static CAC threshold is used by the gateways with the case when the gateways use the continuous CAC threshold adaptation. The utilisation is measured for the link between Node2-Node6. The benefits of the continuous adaptation technique are clear, especially in high load situations. The percentage of calls transferred with noticeable speech degradation decreases by more then 20%, from ~27% to less than 5%.

Table 3.11: The effect of the continuous adaptation method in the multihop simulation

	Blocking	**1%<**	**3%<**	**10%<**	**Util.**
Without adaptation	0.2194	0.2121	0.0205	0.001	1344909
With adaptation	0.3413	0.0562	0.0043	0.0002	1223361

Summary

The end–to–end measurement based call admission control method was studied with extensive simulations, and it was demonstrated that it is capable of ensuring the required low packet loss ratio for telephony calls, while maintaining reasonable link utilization. Configuring the CAC decision very differently in IP telephony gateways, which share a common bottleneck leads to unfair share of the bandwidth. This is an undesirable property but it is a general problem of all endpoint based congestion and admission control solutions[2]. The simulation results revealed the drawback of basic end–to–end measurement based admission method when it comes to supporting wide range of network loads. As a solution, two enhancement of the method were proposed, both of them relying on dynamically adapting the CAC threshold as a response of increasing level of congestion.

The proposed solution is best applicable in scenarios where the bandwith is not too expensive so slightly lower bottleneck link utilisation can be tolerated and in case when the bottleneck bandwidth is by a couple of orders of magnitude higher then the bandwidth requirement of individual calls.

The results presented here demonstrate that end–to–end measurement based call admission control is a viable option for IP telephony gateways, and it represents a very reasonable trade–off between the complexity of the method, and the offered performance guarantees.

3.3.4 End-to-end Measurement Based Call Admission Control in a Best Effort IP Network

Motivating Example

Recent measurement results show that the Internet is dominated by TCP traffic [53], but the fraction of real–time multimedia content is steadily increasing [54]. Floyd and Fall [55] consider the potentially negative impacts of an increasing deployment of non–congestion–controlled traffic over the Internet. It concludes that all application (including real–time multimedia services) deployed over the Internet should use some sort of end–to–end congestion control. My research was focused on investigating to what extent can

[2]The reader is referred to the well–known issue of TCP-friendlyness: http://www.psc.edu/networking/tcp_friendly.html.

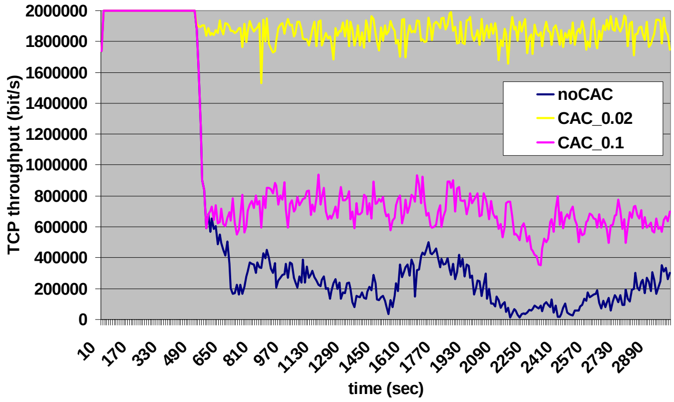

Figure 3.19: TCP throughput

end–to–end measurement based call admission control be used as a replacement for end–to–end congestion control when the task is to protect legacy TCP flows from unregulated real–time traffic but also provide acceptable quality for real–time sessions.

The simulation setup contains 2 routers. They are interconnected by a 2 Mb/s link. The first node initiates the voice calls, and the second one gathers only loss statistics and reports it back. The average flow inter–arrival time is 0.4 second. The length of the simulation is 3000 seconds. Three FTP sources start at 1.0 second of the simulation time. 500 seconds later ON/OFF calls start to flood the network. In the first case, the ON/OFF sources were not regulated. In the second and third case they are subject to CAC decision before entering the network. The call admission control method takes only the loss measurement into account, and the threshold is 0.02 in the second case, and 0.1 in the third one. Figure 3.19 depicts the aggregated throughput of the 3 TCP sessions over subsequent 10 seconds intervals.

The average utilization of the link is 99% if there is no call admission control, it is 96% if the CAC threshold is 0.02, and it is 90% with CAC threshold of 0.1. The call blocking probability is 95% with the stricter threshold, and drops to 42% with the 0.1 threshold.

If there is no call admission control TCP flows are expelled from the link, which is definitely not a desired phenomenon. We can see that the 0.02 CAC threshold is far too conservative, and admits so few calls that TCP is hardly effected at all. The remainder of this section is dedicated to take a closer look at this phenomenon in many different configurations, revealing not just the throughput of the TCP sessions but also the resulting voice quality.

Effect of the Link Speed and the Number of TCP sessions

The first experiment is run with a single bottleneck link interconnecting two routers. The bottleneck link is shared by varying number of TCP flows and admission controlled voice traffic. The utilisation of the bottleneck link is measured, and the fraction of bandwidth received by the admission controlled traffic is calculated. The measurement period is set to 1s. Figure 3.20 depicts the bandwidth share of the admission controlled flows. The value of 0.5 would represent fair sharing between the TCP flows and the admission controlled traffic. The average flow inter–arrival time of the voice traffic is scaled with the bandwidth of the bottleneck link. It is set to 1.5/linkrate seconds, where the linkrate is measured in units of Mb/s. The loss threshold is set to 0.1 and the delay threshold is set to 150 ms, which corresponds to acceptable one–way delay for real–time voice communication. (Subjective tests reported in Annex B of [45] say that the communication quality is acceptable if the one–way mouth to ear transmission delay stays below 400–500 ms depending on for example the interactivity level of the conversation. Based on this figure we assume in this paper that about 150–250 ms can be consumed from the delay budget in the IP core network.)

We can observe that admission control solution leads to a reasonably fair bandwidth sharing (value between 0.4–0.6) over a large operating region. TCPs are much more aggressive in fighting for bandwidth if a small bottleneck link (1–4 Mb/s) is used by a large number of TCP flows. On the other hand, the admission controlled traffic gets more then its fair share if it competes with only one or two TCP flows. However, even in this extreme case, the TCP traffic is not completely expelled. The utilisation of the bottleneck is very high in all simulated cases. Its minimum is 74% when there is only a single TCP flow competing with the voice traffic over a 32 Mb/s link. The average utilisation over all 42 simulated cases is 96%.

To assess the quality of service guarantees provided for the voice traffic the blocking probability and the fraction of individual voice calls, which suffered more then 10% packet loss are measured, and the 10^{-2} percentile of the delay of the voice packets is also calculated. Figure 3.21 depicts the blocking probabilities, Figure 3.22 shows the loss statistics, and the delay values can be seen in Figure 3.23[3].

The blocking probability graph confirms the observation above that in case of a small bottleneck link and many simultaneous TCP flows, the admission controlled traffic is hit very hardly.

Figure 3.22 and Figure 3.23 demonstrate that not just the overall bandwidth is shared fairly, but also the quality of service experienced by individual voice sessions is satisfactory. The packet loss ratio of all individual calls is below 10% for the larger part of the simulated parameter range. Once

[3]Please note that in order to improve the look of the 3-D charts, the labelling of the axes of the charts is different in the 3 figures.

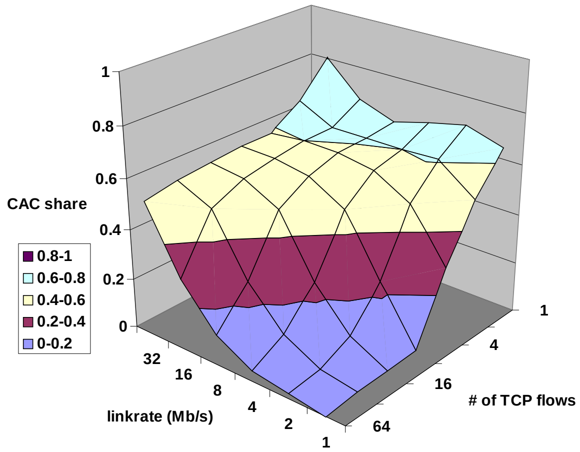

Figure 3.20: Bandwidth share of admission controlled flows

again, one can note that problems start to occur when many TCP flows are competing against the voice traffic over a small link. The worst case queuing delay in this configuration is around 120 ms, which is measured when the small buffer configured for a 1–2 Mb/s link is almost constantly kept occupied by the large number of active TCP flows. The above mentioned worse case delay value also indicates that all the blocked calls are rejected due to high loss values, and never due to excessive queuing delay.

To learn more about the degradation of the communication quality between the two end of the scale measured during this experiment, two more parameters were calculated. The first one is the maximum length of loss bursts that is the maximum number of consecutive packets that has been lost from a particular voice session. This parameter indicates[4] the length of annoying periods of speech clipping due to packet loss. The second parameter is the maximum length of speech burst, which is the maximum number of consecutive speech packets that has been delivered without a single packet loss. This value measures the length of completely undistorted speech periods. Table 3.12 summarises the measurement result. A speech burst of 8918 packets represents approximately 3 minutes of talking without a single packet loss.

Effect of the CAC Threshold

The second experiment concentrates on the effect of different loss thresholds used by the call admission control algorithm. These simulations are also run with a single bottleneck link, which has 8 Mb/s capacity. The delay

[4]Note that this parameter can not be used as a direct measure of the length of the speech clipping period because the error concealment unit of the voice coder can significantly alleviate the user–perceived effect of packet loss.

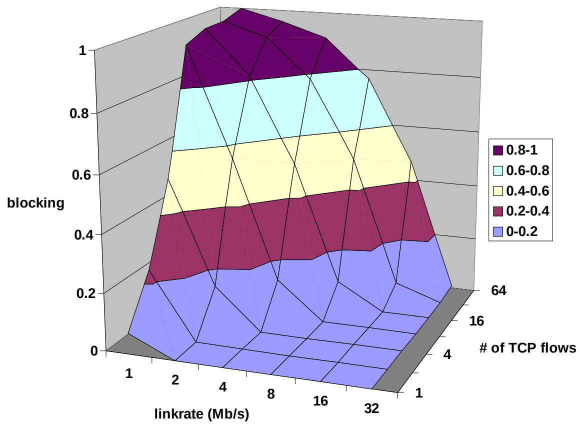

Figure 3.21: Blocking probability

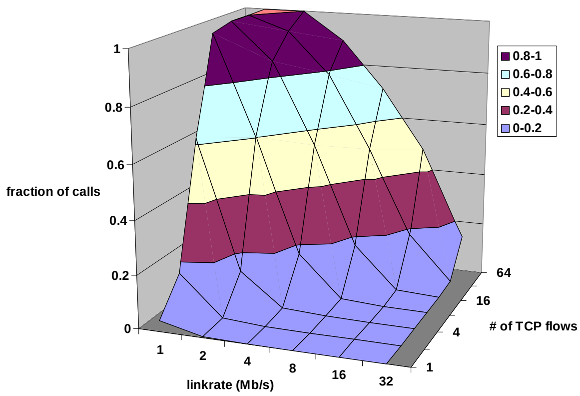

Figure 3.22: Fraction of individual voice calls with more then 10% loss

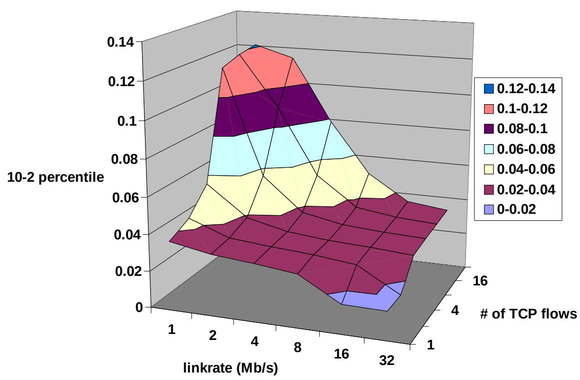

Figure 3.23: Voice packet delay (sec)

Table 3.12: Length of undistorted speech and loss periods

Linkrate	**# of TCPs**	**Loss burst**	**Speech burst**
2 Mb/s	64	9	53
8 Mb/s	8	7	409
32 Mb/s	4	4	8918

measurement results are not taken into account when making the admission control decision. Figure 3.24 depicts the bandwidth share of the admission controlled traffic. The minimum utilisation of the bottleneck is 95%, while the average utilisation over all the simulation runs is higher then 99%. If there is no call admission control, the non–responsive voice traffic dominates the link. By lowering the call admission control threshold, smaller and smaller number of TCP flow is enough to suppress the admission controlled traffic. These curves show that end–to–end measurement based call admission control can ensure fair sharing of the link capacity only in a limited region. One set of simulation is run to check the effect of Random Early Discard (RED). RED is configured as in [44]: the buffer capacity is set to 120, the minimum threshold is 15 packets, the maximum threshold is 50 packets, the maximum drop probability is 0.1 and the weight used in the queue size calculation is 0.002. If there are many parallel TCP sessions, RED buffer management results in noticeably higher blocking probability and lower share for the voice sessions compared to the case when using the same CAC threshold but droptail buffer management. The difference in the bandwidth share is about 8-14%. This observation is in line with the fact that RED helps improving the throughput of simultaneous TCP sessions by avoiding synchronized shrinking of the congestion windows.

Figure 3.25 depicts the blocking probabilities. The amount of traffic generated by the voice flows is proportional to the number of flows admitted into the network, therefore it is not surprising that the blocking probability curves basically mirror the corresponding bandwidth share curves.

Figure 3.26 shows the loss performance of individual voice flows, and Figure 3.27 depicts the voice packet delay. Setting the CAC threshold to the value of 0.07 results in far the best performance. Even if there are 64 competing TCP flows, only 8% of the voice calls suffer more packet loss then the critical 10%. The voice QoS degrades significantly with the other four settings. The worst performance is delivered if the voice calls are not regulated at all. Figure 3.24 and 3.26 demonstrate the benefit of applying even this very simple call admission control technique. The proposed method can definitely prevent the worst possible scenario, when non–adaptive voice flows flood the bottleneck link, and they not only expel TCP traffic from the network, but render also the voice flows to unacceptable performance. It is interesting to see the effect of RED buffer management on the packet

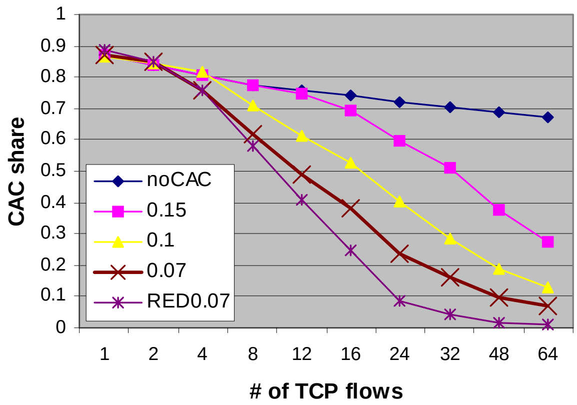

Figure 3.24: Bandwidth share of admission controlled flows

loss suffered by individual voice flows. RED starts to make a very big difference when the number of TCP flows is high. Random drops help TCPs maintaining larger congestion window, which result in larger portion of the buffer being allocated by TCP packets, so higher chance for a voice packet to arrive to the buffer when the queue length is measured to be above the maximum threshold.

All 4 CAC thresholds yield roughly the same delay performance if the number of TCP flows is below 12 and the queue management method is tail drop. In case of RED queue management, the delay performance is better because in a lightly loaded system RED drops more packets then tail drop, but there is not enough TCP flow to benefit from random drops, so the buffer occupancy, and consequently the voice packet delay is smaller. In this region the number of TCP flows is not really effecting the delay, the curves are flat. As the number of TCP flows go beyond 12, the different loss thresholds yield to significantly different delay performance. Stricter loss threshold means less and less voice flows in the system when the number of competing TCPs increase. This means that the portion of large TCP packets in the system increases therefore the voice packets are queued behind higher and higher number of 1000 bytes long TCP packets, which explains the difference between the delay curves.

Effect of Measurement Interval

The third experiment tries to answer the question whether higher intensity of feedback messages makes the admission controlled traffic more competitive compared to TCP. These simulations are also run with a single bottleneck link, which has 8 Mb/s capacity. 3 simulations are run with a delay threshold of 80ms. The voice packet delay is averaged over the measurement interval, and new calls are blocked if the average delay exceeds the 80 ms

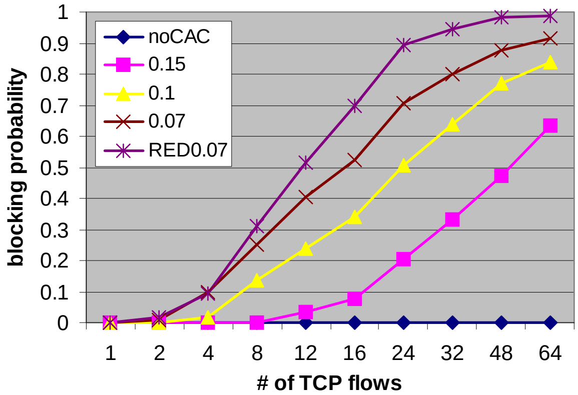

Figure 3.25: Blocking probability

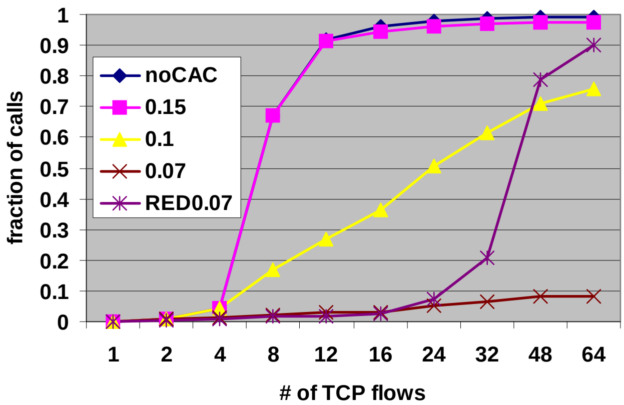

Figure 3.26: Fraction of individual voice calls with more then 10% loss

delay threshold. Figure 3.28 shows the bandwidth share of the admission controlled traffic and Figure 3.29 depicts the loss performance of individual calls.

When the background load is light (1-16 TCP flows), there is a benefit of configuring more frequent measurement reports. (0.1s or 0.5s instead of 1.0s). In this case the reason for call blocking is mainly packet loss, which is confirmed by the fact that the performance of the method is the same even if no delay threshold is configured (mea1.0_nod). As a reference, the graphs also show the measurement results for the case when the call admission control method is switched off (noCAC). The utilisation of the bottleneck link is above 99% if averaged over all the simulation runs. Increasing the frequency of measurement reports by a factor of 10 results in a smaller bandwidth share for the voice traffic. (The difference is approximately 3%.) As a reward, the loss performance of individual calls get better that is the fraction of calls delivering unacceptable performance decreases by approximately 10%.

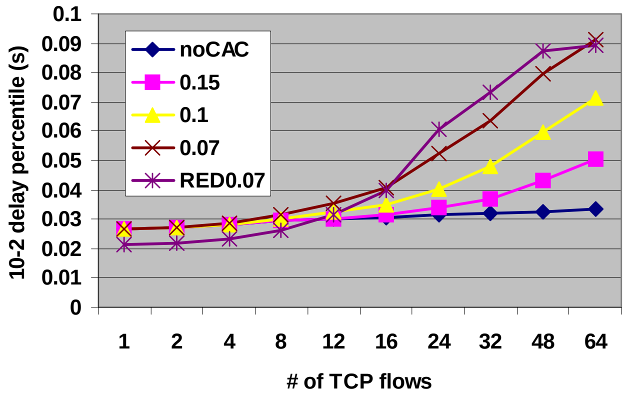

Figure 3.27: Voice packet delay

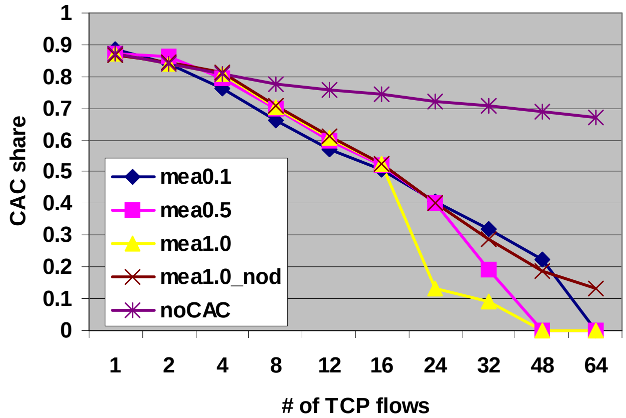

Figure 3.28: Bandwidth share of admission controlled flows

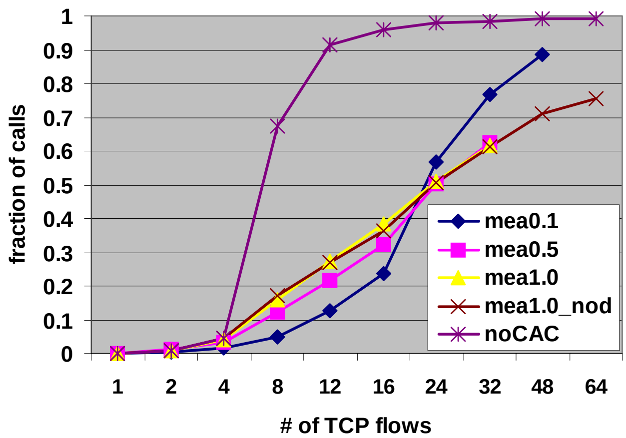

Figure 3.29: Fraction of individual voice calls with more then 10% loss (3 curves are incomplete due to blocking of all calls in that particular setting)

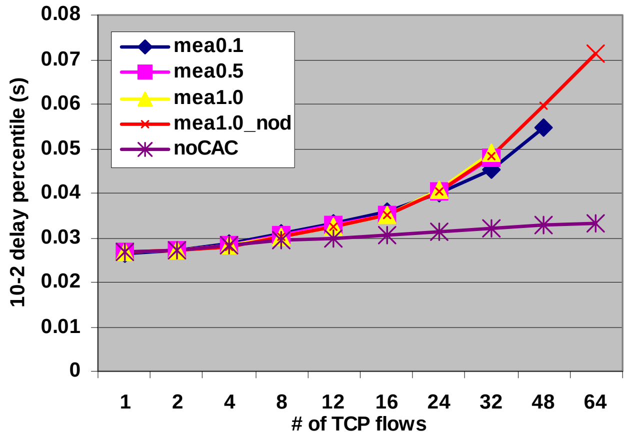

Figure 3.30: Voice packet delay (3 curves are incomplete due to blocking of all calls in that particular setting)

On the other end of the scale when there are a lot of TCP flows, blocking also occurs due to measuring higher average delay then the delay threshold. The method with more frequent feedback is less conservative. The benefit is however questionable since the per–flow loss values are very bad. Figure 3.30 demonstrates that the delay for 99% percent of the voice packets is well below the delay threshold in all settings.

Figure 3.30 gives the impression that switching on the use of delay measurements in the call admission control method makes the method more conservative when it comes to fighting for bandwidth. The aim of the final set of simulations is to justify this observation.

A new network topology, depicted in Figure 3.31 is used in the simulations. The capacity of the two backbone links (node2–node3, node3–node6) is 8 Mb/s, the capacity of the other links is 100 Mb/s. The end–to–end measurement based admission control method operates between node1 and node4. The link between node2 and node3 is a bottleneck during the whole simulation, because TCP flows generate traffic from node0 to node5. Between 650 and 800 second of the simulation time, there is also TCP traffic generated between node8 and node7. The number of simultaneous TCPs is equal on both bottlenecks. The measurement results are available in Table 3.13. Three simulations are presented. The first simulations are run with a single bottleneck, the feedback interval is 0.1s, but the delay and loss thresholds are different. The third one is run with two bottlenecks without exploiting the delay measurements in the CAC algorithm.

Comparing the first and second column of Table 3.13 reveals that a smaller delay threshold leads to a very abrupt change in bandwidth share of the admission controlled traffic, when the number of competing TCP flows increases. Once there is enough TCP in the network to raise the bottleneck

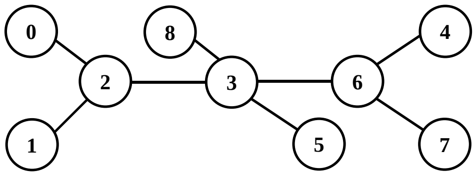

Figure 3.31: Network configuration with two bottleneck links

Table 3.13: Summary of the simulations with different delay thresholds

1 bottleneck, meas=0.1s, delay<0.05s, loss<0.1			
# TCPs	**CAC share**	**>10% loss**	**Blocking**
1	0.888	0.002	0.004
8	0.443	0.054	0.447
16	0.341	0.219	0.598
24	0	N/A	1
32	0	N/A	1
1 bottleneck, meas=0.1s, delay<0.08s, loss<0.07			
# TCPs	**CAC share**	**>10% loss**	**Blocking**
1	0.854	0.002	0.016
8	0.588	0.015	0.299
16	0.403	0.03	0.492
24	0.289	0.134	0.647
32	0.222	0.229	0.723
2 bottlenecks, meas=1s, no delay th., loss<0.1			
# TCPs	**CAC share**	**>10% loss**	**Blocking**
1	0.897	0.001	0
8	0.762	0.053	0.043
16	0.6	0.244	0.244
24	0.465	0.367	0.371
32	0.386	0.457	0.467

buffer occupancy above a limit corresponding to the delay threshold of the admission control algorithm, there will be no new call admitted. Setting the delay threshold to a value, which is smaller than the maximum queuing delay basically means that voice packets can only be queued up to a buffer limit corresponding to the preset delay threshold, while TCP packets can exploit the whole buffer. This inevitably leads to expelling of the voice traffic earlier compared to the case when no delay threshold is set in the CAC method.

Summary

The method was studied with extensive simulations. The goal of the investigation was to understand the effect of TCP flows on the admission controlled traffic. The most important findings of the section are as follows.

Increasing the number of TCP flows inevitably results in complete expelling of the admission controlled traffic, simply because TCP is not that concerned with the packet loss ratio, and can work nicely with much higher packet loss ratio then the one which can be tolerated by the voice traffic. The method however has a very definite benefit:

- It blocks all those voice calls, which would anyway be useless because of the very high packet loss ratio, and in this way ensures much higher throughput for TCP traffic.

In the range, where the number of TCP flows present in the network is modest, that is the resulting packet loss ratio is not more then the one which can be tolerated by the voice traffic, the method is extremely beneficial for two reasons:

- It protects TCP traffic from non–responsive flows, and result in an approximately fair share of the bandwidth between the two traffic classes.

- It ensures the required packet loss ratio for the voice flows.

The method can ensure that the delay bound of the voice packets is not exceeded. Taking however delay statistics also into account in the admission control decision makes the call admission control method even more conservative when fighting for bandwidth against TCP flows.

3.3.5 The Effect of Packet Loss on the Feedback Path

As it is clear from Section 3.2, the receipt of regular feedback messages is crucial for the proposed CAC method to work. From [41] we know that in an operational IP network packet loss ranges between 0% and well over 10% can be measured depending also on the length of measurement interval, but more then 10% packet loss ratio was measured only in less then 1% of the traces. The loss rate in the two direction is often different, but the order of magnitude in the two directions is fairly symmetric.

The aim of the investigation in this section is to assess the effect of update packet losses on the performance of the proposed CAC solution. If update packets are lost, the gateway can not update its estimate of current conditions in the network which can potentially lead to two undesirable effect:

- The gateway is stuck in the state of admitting calls which can lead to severe overload.

- The gateway is stuck in the state of rejecting calls which can lead to high blocking probability and underutilisation of network resources.

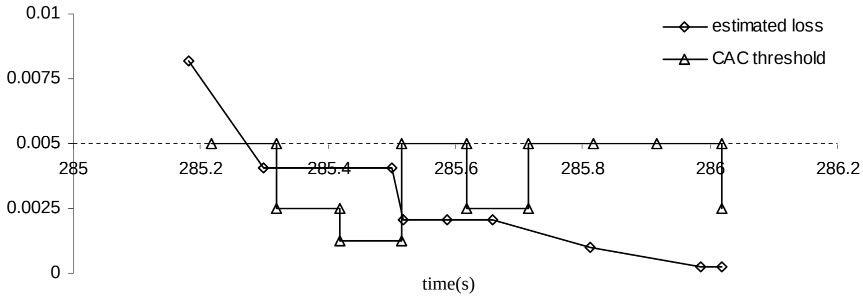

Figure 3.32: Evolution of the loss estimation and the CAC threshold

To protect the system against the potential negative effect of lost update packets, a timer is introduced to supervise the receipt of update packets. This timer is restarted upon receipt of a new update packet. If the timer expires, the admission control threshold is lowered, more precisely it is divided by a factor called *backoff*. This procedure ensures that the admission control is made stricter. If an update packet is finally received, the call admission control threshold is reset to the initial value.

I experimented with backoff parameter values 2 and 4 using measurement intervals between 0.1s and 10s. The packet loss ratio over the feedback path was set to 0%, 15%, 30% and 45%. The simulations are run using the simplest topology: a single link interconnecting two routers. The link speed was 2 Mb/s, the propagation delay was 15ms, and the average flow interarrival time was 0.3s. The initial value of the call acceptance threshold was 0.005. Delay measurements were not utilised, the feedback packets reported only about packet loss. Figure 3.32 depicts an extract from the simulation results clarifying the operation of the enhanced method. Triangles show the evolution of the CAC threshold while rhombuses mark the estimated loss calculated from the update packets. If the curve of triangles is above the other curve then new calls are accepted, otherwise they are rejected.

As it can be expected, the loss of update packets negatively effects the performance of the admission control algorithm. The degree of degradation is depicted in Figure 3.33 where the fraction of calls suffering more then 1% packet loss is shown.

The 0.1s measurement interval performs uniformly bad regardless of the update packet loss ratio. More then 40% of the calls lost more than 1% of their packets. The high loss ratio is caused by the short measurement interval. As the number of sources currently in off period change, the link may seem unexploited in a short term. This results in more admitted flows, which leads to congestion when sources switch to on period. 1s is the optimal measurement interval (it may depend on the link speed) when the damage made by the loss of update packets is also minimal. Above the 1s measurement interval the performance of the method degrades. Infrequent updates

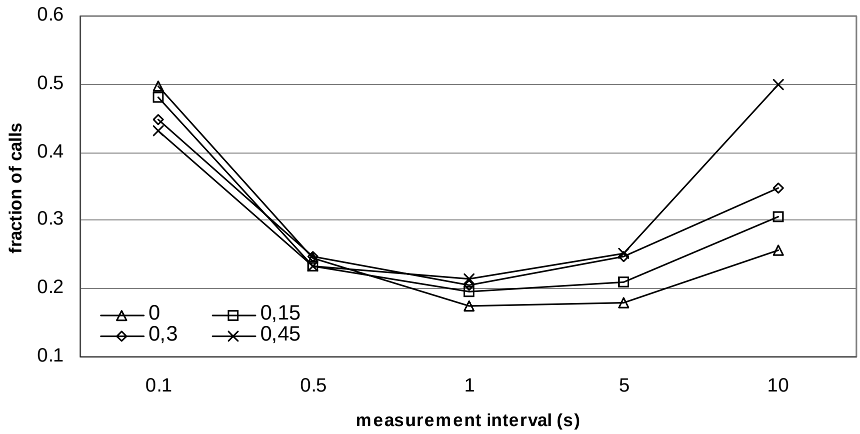

Figure 3.33: Fraction of calls suffering more then 1% packet loss when update packets are lost

may cause long accept periods and thus congestion, or long reject periods and unutilized resources. The loss of feedback packets worsen further the performance by dropping some of the already few updates. A noticeable decrease of link utilization also proves that in Figure 3.34.

Figure 3.35 demonstrates the effect of different backoff values. The curves meet where update loss ratio is 0, since the lack of losses makes the backoff mechanism unnecessary. The benefit of the proposed enhancement is clearly visible on the figure. The backoff mechanism improves the loss performance in each setting. The improvement is most noticeable when the update loss ratio is high (0.45). The fraction of calls suffering more then 1% packet loss is halved by setting the backoff value to 4. In return for good results regarding packet loss, the link utilization is usually lower with the backoff technique. Note that increasing update packet loss ratio opens up the gap between the curves, the backoff mechanism becomes more and more useful. Without the backoff mechanism, the fraction of calls experiencing more then 1% packet loss increases when update packet loss increases, while the backoff mechanism results in lower number of badly effected calls. Unfortunately, link utilization also decreases, due to the stricter admission decision upon update loss, but in most cases providing QoS is more important than utilisation especially when it comes to supporting a telephony–like service. Backoff value of 4 achieved outstanding results with measurement interval 0.5s and 1s when the update packet loss ratio was 0.45. In the first case the fraction of calls with more than 1% packet loss was only 8.6%, while in the second case it was only 7.8% combined with high link utilization (85% and 83%). These simulations proved that a busy network with unprecedented high update packet loss ratio can cause performance problems for the end–to–end measurement based call admission control, but a

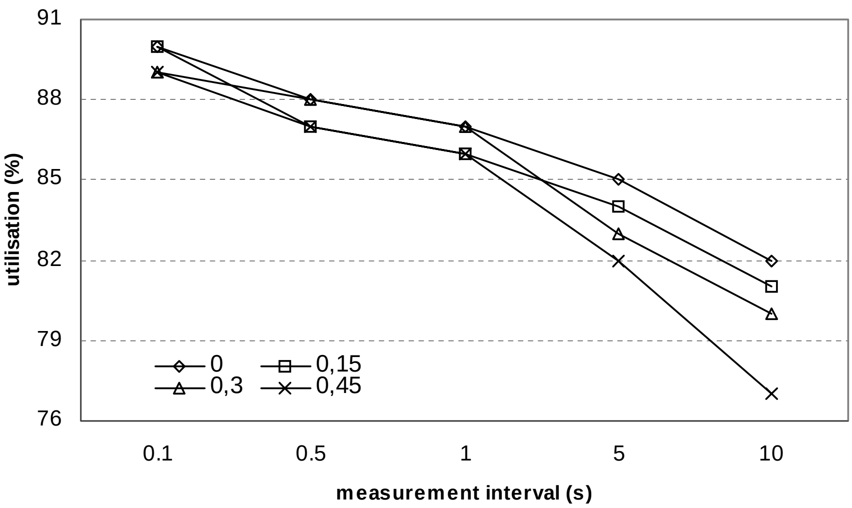

Figure 3.34: The effect of update packet loss on the utilization

simple enhance can factor this effect also into the usual equation of trading off QoS guarantees for utilisation.

3.4 Applicability of the Proposed Method in VoIP Terminals

The solution presented in Section 3.2 exploits the fact that there are hundreds of simultaneous voice calls between any two gateway pairs, therefore measuring the quality of ongoing live sessions provides a reasonably accurate picture about the current state of the network. If however a VoIP terminal is about to start a new session, it does not have the necessary statistics to base the call admission control decision on. In this section I outline a measurement architecture to allow the use of the end–to–end measurement based call admission control technique by individual terminals.

3.4.1 Clustering of Clients Using BGP Routing Information

This subsection presents a technique for partitioning of a set of IP addresses into non-overlapping groups, where all the IP addresses in a group are topologically close and under common administrative control. Such a group is called a *cluster* and I use these clusters as a basis of a measurement infrastructure enabling the extension of the end–to–end measurement based admission control technique to VoIP terminals. The method was proposed in [56] but it was used there in a completely different application scenario: they used the solution to cluster clients of a web server in order to determine the best location for deploying new proxy servers.

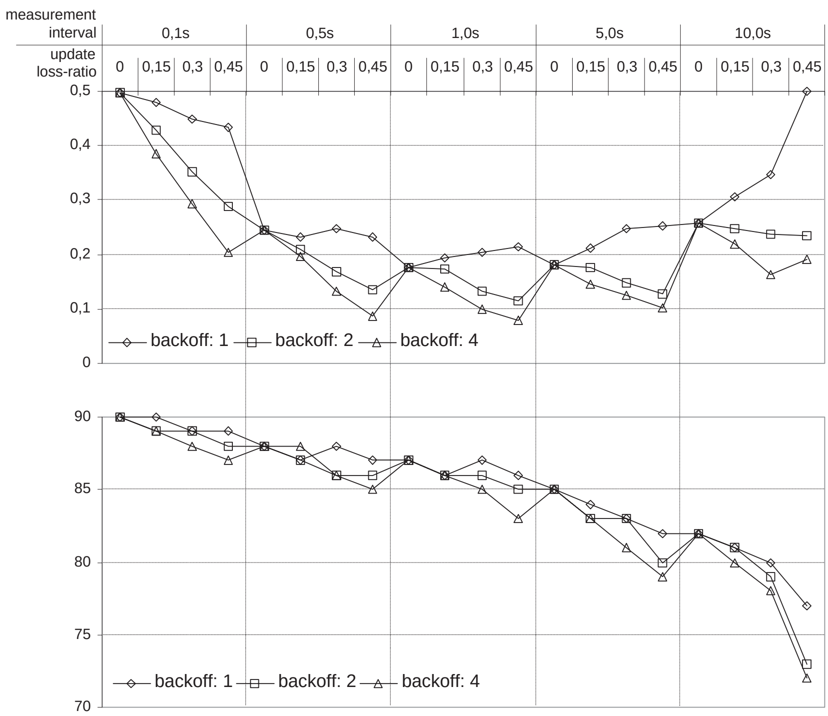

Figure 3.35: Fraction of calls suffering more than 1% packet loss and link utilization, as a function of backoff values

The clustering method uses the prefixes and netmasks extracted from the BGP (Border Gateway Protocol) [57] routing and forwarding table snapshots. The rationale behind the approach is that the prefixes and corresponding netmasks identify the routes in the routing table which are used by core routers to forward packets to a given destination. The most specific entries are most useful, since they represent groups of clients that are topologically very close together. By examining multiple routing tables, it is likely to get a large number of specific entries that include most of the client IP addresses.

The process is composed of five steps: (i) network prefix/netmask extraction and extraction of IP addresses from Web server logs; (ii) client cluster identification; (iii) (optional) validation of sampling the resulting client clusters; (iv) examining effect of network dynamics on client cluster identification; and (v) self–correction and adaptation.

The first step is to generate a combined prefix/netmask entry table. It is done by extracting prefix/netmask entries from the routing table snapshots, unifying the entries into a standard format, and inserting all the entries from different routing tables into a single, large table.

The second step is the actual clustering of clients which involves performing the longest prefix matching (similar to what IP routers do) on each client IP address using the constructed prefix/netmask table and classifying all the client IP addresses that have the same longest matched prefix into one client cluster, which is identified by the shared prefix. Suppose we want to cluster the IP addresses 12.65.147.94, 12.65.147.149, 12.65.146.207, 12.65.144.247, 24.48.3.87, and 24.48.2.166. The longest matched prefixes are respectively 12.65.128.0/19, 12.65.128.0/19, 12.65.128.0/19, 12.65.128.0/19, 24.48.2.0/23, and 24.48.2.0/23. We then classify the first four clients into a client cluster identified by prefix/netmask 12.65.128.0/19 and the last two clients into another client cluster identified by prefix/netmask 24.48.2.0/23.

When it comes to validation, a client cluster may be misidentified by being either too large (i.e., containing clients that are not topologically close or are under different administrative control) or too small (i.e., other clusters contain clients that belong to the cluster in question). The tests presented in [56] show that 99.9% of the clients can be grouped with the proposed method. They also presented two methods for validation of the clusters, the first one using DNS lookup and the second one relying on traceroute. Test results are excellent, 90% of the identified clusters pass the validation test.

An update of BGP routing table is triggered by changes in network reachability and topology, as well as policy changes. BGP dynamics is a well–known phenomenon which affects the performance of Internet applications [58]. Experiments presented in [56] however shows that overall BGP dynamics affects less than 3% of client clusters.

The last issue is self–correction and adaptation. For the purpose of identifying unidentified clients ($\sim 0.1\%$), first each individual unidentified client is considered to be a single client cluster. Then they are merged into bigger client clusters gradually according to traceroute sampling information.

Two cases are considered in the self-correction and adaptation process: (i) if there is more than one cluster which belongs to the same network, we merge them into one big cluster and the network prefix and netmask will be recomputed accordingly; (ii) if there is a cluster which contains clients belonging to more than one network, we partition the cluster into several clusters based on the traceroute sampling results.

Self–correction and adaptation are very important to generate client clusters using real–time routing information and producing real–time client cluster identification results. Important to note that the entire cluster identification process can be done in an automated fashion and the methodology is immune to BGP dynamics and it is computationally non–expensive.

3.4.2 A Distributed Measurement Architecture for Performing Call Admission Control Based on End–to-end Measurements

A new entity called Call Admission Control Broker is introduced which is responsible for collecting network measurement reports from terminals, arranges these reports into a database and it is responsible for making the call admission control decisions upon a new call is initiated by the terminals.

If a VoIP terminal would like to start a new telephone call it sends a Call Request message to the Call Admission Control Broker. The message contains at least the IP address of the calling party and the called party. The broker makes a call admission control decision and the result is reported back to the terminal (Call Accept message). If the decision is positive the terminal can proceed with the voice session. During the voice sessions the terminals monitor the transmission quality. They can collect loss and delay statistics. Right after the termination of the telephony call the terminal reports the measurement results to the Call Admission Control Broker in a Measurement Report message. This message contains the packet loss ratio and the average delay measured during the session and informs the broker also about the IP address of calling and called party.

The Call Admission Control Broker implements the clustering method described in 3.4.1. It classifies all the IP addresses received in Call Request messages into clusters. It maintains the state variables (loss, delay) that are used by the call admission control method for each pair of clusters. Upon arrival of a new Measurement Report message about the call quality measured for a voice session between a particular pair of clusters, the state variables corresponding to the cluster pair are updated. If a Call Request message arrives to the Call Admission Control Broker, the broker checks the IP address of the called and calling party, and determines the clustering of both VoIP terminals. The network state variables corresponding to the cluster pair are used in the call admission control process. The outcome of the call admission control decision is reported back to the VoIP terminal.

Even if the IETF standard Session Initiation Protocol (SIP) [59] is used by the VoIP terminal for establishing the voice session, and the actual signalling takes place end–to–end without the involvement of any intermediate proxy, there is now a way for blocking the call if the required resources can not be set aside for ensuring outstanding voice quality. [60] discusses how network QoS and security establishment can be made a precondition to sessions initiated by SIP, and described by SDP. These preconditions require that the participant reserve network resources before continuing with the session. These preconditions require a participant to use an existing resource reservation mechanisms before beginning the session. This results in a multi–phase call–setup mechanism, with the resource management protocol interleaved between two phases of call signaling. Figure 3.36 depicts an example of integrating the proposed call admission control mechanism into SIP end–to–end session control signalling using the approach of [60]. This chart covers the case of successful session establishment.

3.5 Conclusions

This chapter presented a new approach for making call admission control for telephony sessions in an IP network. The salient feature of the proposed method is that it is fully transparent to the routers in the network core. The method is comprised of different building blocks. These building blocks make possible to put together a call admission control solution which allows the network operator to adapt the method to its networking environment and to select the preferred operating point on the trade–off between network utilisation and performance guarantees. The toolbox includes multiple parameters to base the admission decision on and adapting the operation of the method to reflect changing conditions in the network. The solution is primarily targeted for IP telephony gateways, but it can also be used by individual VoIP terminals. Detailed simulation results prove the viability of the proposal.

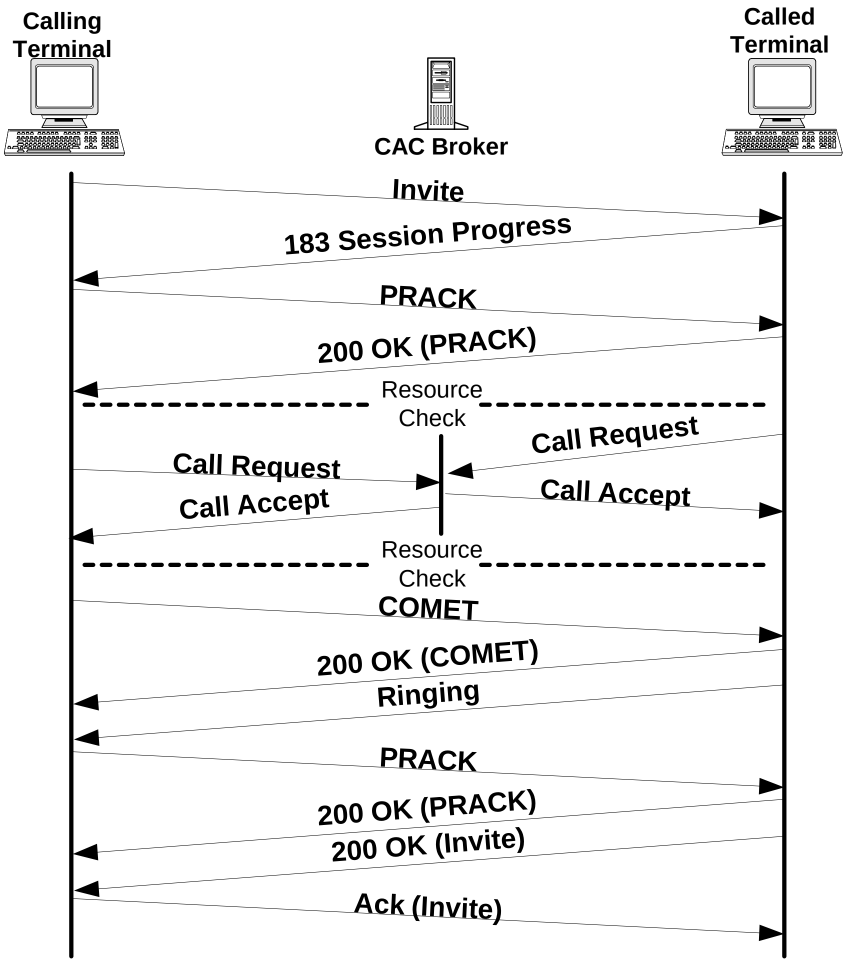

Figure 3.36: Integrating the Resource Checking with SIP Signalling

Chapter 4

Closing Remarks

4.1 Contribution of this Thesis

My research concentrated on two main areas. The first area being signalling in ATM networks. As a result of this research I conceptualised a new signalling protocol for controlling end–to–end, switched AAL2 connections. I introduced the concept of *Bearer Independent Signalling Protocol*, which makes possible for a signalling protocol to operate on top of any underlying bearer network, for example SS7(MTP), IP or ATM. As a proof of concept, I provided two sample design of a so-called *bearer converter* which makes possible carrying the AAL2 Signalling protocol on top of SS#7 network and on top of ATM Signalling Adaptation Layer. Finally, I proposed to implement AAL2 Signalling using priority queuing of the protocol messages and I evaluated the solution using elementary queuing theory and simulation.

The second part of my research led to a new method for performing call admission control in IP Telephony networks. I proposed a core–stateless, end–to–end measurement based call admission control method. The salient feature of the proposal is that it works without putting any burden on the routers in the network core. The original method is best suited for implementation in IP telephony gateways. In addition I proposed a measurement architecture which makes possible using the passive end–to–end measurement based CAC not only in IP telephony gateways but also in VoIP terminals. I have validated the benefits of the proposed call admission control solution by analysing the results of an extensive set of simulations. I have also investigated what happens when per–flow admission control is mixed with flow–level congestion control in an IP network. The results characterise the voice quality, the throughput achievable by TCP traffic as well as the fairness of the resource sharing between the two traffic types, and the effect of congestion on the feedback path.

4.2 Application of New Results

I contributed with the details of the AAL2 Signalling protocol to Study Group 11 of ITU–T [S1]–[S24]. My proposals have been incorporated into new ITU–T Recommendation Q.2630.1 [10] and Q.2150.1 [11]. Q.2630.1 has later been standardised by 3GPP [12] to be used to control AAL2 connections in UMTS Terrestrial Radio Access Networks. [J2] is cited in five independent publications [Cit1]–[Cit5].

I first introduced the concept of *Bearer Independent Signalling Protocol* for AAL2 Signalling, but the idea has later been applied for many new protocols such as the Stream Control Transmission Protocol in IETF [61] and the Bearer Independent Call Control Protocol [62] in ITU–T.

The AAL2 signalling protocol and the corresponding bearer converter I proposed has been implemented by many companies (Ericsson [21], Trillium [22], Spirent Communications [23]), and it is now part of commercial product offerings.

The Internet is on its way to become a universal service platform offering real–time multimedia services as well as simple file transfer, web browsing and e–mail. However, for the time being most of the routers deployed in the network are capable of providing simple best effort packet forwarding, and it will take significant time and money to upgrade these routers to support sophisticated QoS differentiation. There is a need for a sound migration path. In this transition phase there is a definite window of opportunity for solutions which offer some sort of QoS guarantees, therefore represent a step forward from the best effort service paradigm, but operate without requiring any involvement from core routers. The method I propose in Chapter 3 can be realized by upgrading only the IP telephony gateways, and can still provide reasonable QoS for voice sessions under wide range of network conditions.

4.3 Future Work

We are planning a prototype implementation of the end–to–end measurement based call admission control technique in order to enable testing it in a real, operating network.

As mentioned in Section 3.3 there are two fundamental ways of controlling the amount of traffic injected to a network. The first approach is the one promoted by TCP that is to use a congestion control algorithm [63]. The second approach is to handle sessions or traffic flows and perform admission control to limit the amount of traffic. The first approach is used by basically all the data traffic across the Internet, while the second approach and the corresponding notion of session blocking is a property of incumbent telecommunication networks. With the increase of real time traffic over the Internet many researcher promote the introduction of TCP friendly congestion control algorithms for real–time flows in order to improve the communication quality and to ensure fair sharing of resources between the traffic classes [44, 64]. This dissertation investigated a method which belongs to the second approach. Both approaches have drawbacks. An interesting study item is to investigate whether it is possible to integrate these two approaches that is to deploy end–to–end congestion control as well as session level admission control for the real–time flows.

As mentioned earlier, IP will also be used as a transport technology in the core network of 3G mobile systems. There will be an IP tunnel established between the mobile network nodes (SGSN and GGSN) over the IP network upon arrival of a new packet data session. The proposed call admission control solution can be used to permit the establishment of the IP tunnels. Investigating this scenario where the traffic subjected to the call admission control is not homogeneous voice is also an interesting item for future study.

Chapter 5

A Quick Reference Guide to the Theses

Thesis 1 : A Signalling Protocol for Supporting Switched AAL type 2 Connections in UMTS Terrestrial Radio Access Networks

Thesis 1.1 : AAL2 Signalling Protocol [C1, J2, J3]

Section 2.2

Thesis 1.2 : Bearer Independent Signalling Protocol Architecture [P1, C1]

Section 2.3

Thesis 1.3 : Optimising the Performance of AAL2 Signalling [C2]

Section 2.4

Thesis 2 : A Core–stateless End–to–end Measurement Based Call Admission Control Method for Supporting IP Telephony

Thesis 2.1 : Call Admission Control for IP Telephony Based on Passive End–to–end Measurements [C3, P2]

Section 3.2

Thesis 2.2 : A Distributed Measurement Architecture for Performing Call Admission Control Based on End–to-end Measurements

Section 3.4.2

Thesis 3 : Performance Evaluation of the End–to–end measurement Based Call Admission Control Solution

Thesis 3.1 : Performance Evaluation Assuming a Diffserv Capable IP Network [C3]

Section 3.3.3

Thesis 3.2 : On the Interaction of End–to–end Measurement Based Call Admission Control with TCP Traffic [J1, C4]

Section 3.3.4 and Section 3.3.5

Bibliography

[1] *Introduction to 3G Mobile Communications.* Artech House, Norwood, USA, 2001.

[2] *Third Generation Mobile Systems, http://www.ericsson.com/3G.*

[3] *Wideband CDMA for Third Generation Mobile Communications.* Artech House, Norwood, USA, 1998.

[4] IETF RFC 2210. *The Use of RSVP with IETF Integrated Services*, 1997.

[5] IETF RFC 2475. *An Architecture for Differentiated Services*, 1998.

[6] Internet draft, Work in progress. *Load Control of Real Time Traffic*, 1999.

[7] Viktória Elek, Gunnar Karlsson, and Robert Rönngren. Admission control based on end–to–end measurements. In *IEEE Infocom*, Tel-Aviv, Israel, April 2000.

[8] Ion Stoica and Hui Zhang. Providing guaranteed services without per flow management. In *ACM Sigcomm*, Cambridge, USA, August 1999.

[9] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4), November 1984.

[10] ITU–T Recommendation Q.2630.1. *AAL type 2 signalling protocol (Capability Set 1)*, 1999.

[11] ITU–T Recommendation Q.2150.1. *Signalling Transport Converter on MTP3 and MTP3b*, 2001.

[12] 3GPP Technical Specification TR 25.426. *UTRAN Iur and Iub Interface Data Transport & Transport Signalling for DCH Data Streams*, 1999.

[13] ITU–T Recommendation I.363.2. *B-ISDN ATM Adatpation Layer Type 2*, 1997.

[14] Göran Eneroth and Martin Johnsson. ATM transport in cellular networks. In *International Switching Symposium (ISS'97)*, Toronto, Canada, September 1997.

[15] John H. Baldwin, Behram H. Bharucha, Bharat T. Doshi, Subrahmanyam Dravida, and Sanjiv Nanda. AAL–2 — a new ATM Adaptation Layer for small packet encapsulation and multiplexing. *Bell Labs Technical Journal*, April 1997.

[16] Daniel Wong and Teng J. Lim. Soft handoffs in CDMA mobile systems. *IEEE Personal Communications*, December 1997.

[17] Hiroshi Nakamura, Hisakazu Tsuboya, Masatomo Nakano, and Akihisa Nakajima. Applying ATM to mobile infrastructure networks. *IEEE Communications*, January 1998.

[18] ETSI EN 301 816-2 V1.1.1. *Broadband Integrated Services Digital Network (B-ISDN); AAL Type 2 Signalling Protocol; Capability Set 1; Part 2: Protocol Implementation Conformance Statement (PICS) proforma specification*, 2001.

[19] ETSI EN 301 816-3 V1.1.2. *Broadband Integrated Services Digital Network (B-ISDN); AAL Type 2 Signalling Protocol; Capability Set 1; Part 3: Test Suite Structure and Test Purposes (TSS&TP) specification*, 2001.

[20] ETSI TR 301 816-4 V4.0.0. *Broadband Integrated Services Digital Network (B-ISDN); AAL Type 2 Signalling Protocol; Capability Set 1; Part 4: Abstract Test Suite (ATS) and partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma specification*, 2002.

[21] Jonas Reinius. Cello — an ATM transport and control platform. *Ericsson Review*, (2), 1999.

[22] *Trillium's AAL2 Solutions, http://www.trillium.com/products/broadband/aal2/solutions.*

[23] *Extreme AAL 2 Emulation Option, http://adtech.spirentcom.com/datasheets/T4159.asp.*

[24] Ericsson Internal Technical Report, 317/152 83 HSD 108 01/A3. *Test Report Conformance Test (Q.AAL2)*, 2002.

[25] *Signalling System #7.* McGraw–Hill, New York, USA, 1995.

[26] ITU-T Recommendation Q.2130. *B–ISDN Signalling ATM Adaptation Layer — Service Specific Co–ordination Function for Support of Signalling at the User Network Interface (SSCF at UNI)*, 1994.

[27] *Queuing Systems*. John Wiley and Sons, New York, USA, 1976.

[28] 3GPP Technical Specification TR 25.832. *Manifestation of Handover and SRNS Relocation*, 1999.

[29] IETF RFC 2205. *Resource ReServation Protocol (RSVP)*, 1997.

[30] Giuseppe Bianchi, Antonio Capone, and Chiara Petrioli. Throughput analysis of end–to–end measurement–based admission control in IP. In *IEEE Infocom*, Israel, April 2000.

[31] David D. Clark, Scott Shenker, and Lixia Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *ACM Sigcomm*, Baltimore, USA, 1992.

[32] Zoltán Turányi, András Veres, and András Oláh. A family of measurement-based admission control algorithms. In *IFIP Conference on Performance of Information and Communication Systems*, Lund, Sweden, May 1998.

[33] Lee Breslau, Sugih Jamin, and Scott Shenker. Comments on the performance of measurement–based admission control algorithms. In *IEEE Infocom*, Tel-Aviv, Israel, April 2000.

[34] John T. Lewis, Raymond Russell, Fergal Toomey, Brian McGurk, Simon Crosby, and Ian Leslie. Practical connection admission control for ATM networks based on on-line measurements. *Computer Communications*, 21(17), November 1998.

[35] András Veres and Zoltán Turányi. On the tradeoff between efficiency and scalability of measurement-based admission control. In *International Conference on the QoS and Performance of Next Generation Networking*, Nagoya, Japan, November 2000.

[36] Sugih Jamin, Peter B. Danzig, Scott Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated services packet networks. In *ACM Sigcomm*, Cambridge, USA, September 1995.

[37] Richard Gibbens and Frank Kelly. Measurement-based connection admission control. In *International Teletraffic Congress*, Washington, USA, June 1997.

[38] Erik Ekudden, Stefan Bruhn, and Patrik Sörqvist. The adaptive multi–rate speech coder - the new flexible world–standard for voice compression. In *XVII. World Telecommunications Congress*, Birmingham, Great Britain, May 2000.

[39] IETF RFC 1889. *RTP: A Transport Protocol for Real–Time Applications*, 1996.

[40] Yin Zhang, Vern Paxson, and Scott Schenker. The stationarity of internet path properties: Routing, loss, and throughput. Technical report, AT&T Center for Internet Research at ICSI, 2000.

[41] Yin Zhang, Nick Duffield, Vern Paxson, and Scott Schenker. On the constancy of internet path properties. In *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, USA, November 2001.

[42] *"traceroute" can be found at: ftp://ftp.ee.lbl.gov/traceroute.tar.Z.*

[43] Vern Paxson, Jamshid Mahdavi, Andrew Adams, and Matt Mathis. An architecture for large–scale internet measurement. *IEEE Communications*, 36(8), August 1998.

[44] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation–based congestion control for unicast applications. In *ACM Sigcomm*, Stockholm, Sweden, August 2000.

[45] ITU-T Recommendation G.114. *General Characteristics of International Telephone Connections and International Telephone Circuits — One–way Transmission Time*, 1996.

[46] *UCB/LBNL/VINT Network Simulator ns (version 2) can be found at: http://www-mash.cs.berkeley.edu/ns/ns.html.*

[47] Matthias Grossglauser and David N. C. Tse. A framework for robust measurement-based admission control. *IEEE/ACM Transactions on Networking*, 7(3), June 1999.

[48] Coskun Cetinkaya, Vikram Kanodia, and Edward W. Knightly. Scalable services via egress admission control. *IEEE Transactions on Multimedia*, 3(1), March 2001.

[49] Frank Kelly, Peter Key, and Stan Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 18(12), December 2000.

[50] Lee Breslau, Edward W. Knigthly, Scott Shenker, Ion Stoica, and Hui Zhang. Endpoint admission control: Architectural issues and performance. In *ACM Sigcomm*, Stockholm, Sweden, August 2000.

[51] Shuang Deng. Traffic characteristics of packet voice. In *IEEE International Conference on Communications*, Seattle, USA, June 1995.

[52] IETF RFC 2598. *An Expedited Forwarding PHB*, 1999.

[53] Kevin Thompson, Gregory J. Miller, and Rick Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, 11(6), November/December 1997.

[54] Art Mena and John Heidermann. An empirical study of real audio traffic. In *IEEE Infocom*, Tel-Aviv, Israel, April 2000.

[55] Sally Floyd and Kevin Fall. Promoting the use of end–to–end congestion control in the internet. *IEEE/ACM Transactions on Networking*, August 1999.

[56] Balachander Krishnamurthy and Jia Wang. On network–aware clustering of web clients. In *ACM Sigcomm*, Stockholm, Sweden, August 2000.

[57] IETF RFC 1163. *A Border Gateway Protocol*, 1990.

[58] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet routing instability. In *ACM Sigcomm*, Cannes, France, September 1997.

[59] IETF RFC 2543. *SIP: Session Initiation Protocol*, 1999.

[60] Internet draft, Work in progress. *Integration of Resource Management and SIP*, 2001.

[61] IETF RFC 2719. *Framework Architecture for Signaling Transport*, 1999.

[62] ITU–T Recommendation Q.1901. *Bearer independent call control protocol*, 2000.

[63] Van Jacobson. Congestion avoidance and control. *ACM Computer Communication Review*, 18(4), August 1988.

[64] Dorgham Sisalem and Adam Wolisz. Constrained TCP–friendly congestion control for multimedia communication. In *COST Workshop on Quality of Future Internet Services*, Coimbra, Portugal, September 2001.

Publications

Journal Papers

[J1] I. Szabó, *On Call Admission Control for IP Telephony in Best Effort Networks*, Computer Communications, Vol. 26, No. 4, pp. 304-313, March 2003.

[J2] G. Eneroth, G. Fodor, G. Leijonhufvud, A. Rácz, I. Szabó, *Applying ATM/AAL2 as a Switching Technology for 3^{rd} Generation Mobile Networks*, IEEE Communications Magazine, Vol. 37, No. 6, pp. 112-122, June 1999.

[J3] A. Rácz, I. Szabó, *Az ATM/AAL2 alkalmazása UMTS hálózatokban*, Magyar Távközlés, pp. 9-14, Február 2000.

[J4] I. Cselényi, R. Szabó, I. Szabó, N. Björkman, A. Latour-Henner, C. Gisgård, *Experimental Platform for Telecommunication Resource Management*, Computer Communications, Vol. 21 pp. 1624-1640, November 1998.

[J5] R. Ozvald, I. Szabó, *Jelölő algoritmusok teljesítményvizsgálata differenciált szolgáltatást nyújtó IP–hálózatokban*, Híradástechnika, pp. 37-40, Április 2001.

Conference Papers

[C1] I. Szabó, *A Flexible Signalling Protocol for Supporting Switched AAL type 2 Connections in UMTS Terrestrial Radio Access Networks*, IEEE International Conference on ATM, Colmar, France, June 1999.

[C2] I. Szabó, S. Székely, I. Moldován, *Performance Optimisation of AAL2 Signalling for Supporting Soft Handoffs in UMTS Terrestrial Radio Access Networks*, IEEE International Symposium on Computers and Communications, Antibes, France, July 2000.

[C3] I. Szabó, *Performance Evaluation of a New End-to-end Measurement Based Call Admission Control Scheme for Supporting IP Telephony*, SCS, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Orlando, Florida, July 2001.

[C4] I. Szabó, *A Call Admission Control Method for Supporting Telephony Sessions in a Best Effort IP Network*, COST Workshop on Quality of Future Internet Services, Coimbra, Portugal, September 2001.

[C5] I. Cselényi, C. Gisgård, P. Haraszti, A. Latour-Henner, I. Szabó, G. Záruba, *Multipurpose Middleware for Broadband Multimedia Applications*, ConTEL Broadband & Multimedia Workshop, Zagreb, Croatia, November 1996.

[C6] I. Cselényi, I. Szabó, P. Haraszti, N. Björkman, C. Gisgård, *A Versatile Multimedia Platform*, IASTED International Conference on Applied Informatics, Innsbruck, Austria, February 1997.

[C7] B. Benkovics, I. Szabó, *Provisioning QoS in 3G Networks With RSVP Proxy*, International Symposium on Wireless Personal Multimedia Communications, Aalborg, Denmark, September 2001.

[C8] A. Jüttner, I. Szabó, Á. Szentesi, *On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks*, accepted for publication, IEEE Infocom 2003.

Patents

[P1] I. Szabó, G. Eneroth, *Bearer Independent Signalling Protocol*, US Patent and Trademark Office patent application No. 09/064,830, April 1998.

[P2] I. Szabó, *A Lightweight Core–stateless Call Admission Control Method for IP Telephony Gateways*, European Patent Office patent application No. 00850110.8, June 2000.

[P3] I. Szabó, *Supporting QoS Differentiation in RTP Multiplexing Scenarios*, European Patent Office patent application No. 00850111.6, June 2000.

[P4] I. Szabó, *A Method for Improving the Bandwidth Efficiency of RTP Multiplexing while Maintaining Tolerable Delay*, European Patent Office patent application No. 00850112.4, June 2000.

[P5] S. Rácz, Sz. Malomsoky, I. Szabó, *Extending the Content of the Link Characteristics Parameter in AAL2 Signalling to Enable the Use of Much More Reliable Connection Admission Control Methods*, US Patent and Trademark Office patent application No. 10/143,826, May 2002.

Contributions to International Standardisation

[S1] *AAL2 Link Identifier*, Delayed Contribution D.966, ITU-T Study Group 11 Meeting, Geneva, Switzerland, May 1998.

[S2] *Addressing in AAL2 Networks*, Delayed Contribution D.971, ITU-T Study Group 11 Meeting, Geneva, Switzerland, May 1998.

[S3] *Addressing of AAL2 Nodes in a Switched ATM Network*, Delayed Contribution D.972, ITU-T Study Group 11 Meeting, Geneva, Switzerland, May 1998.

[S4] *Making the AAL2 Signaling Protocol Independent of Underlying Signaling Bearer*, Delayed Contribution D.973, ITU-T Study Group 11 Meeting, Geneva, Switzerland, May 1998.

[S5] *AAL2 Signaling Protocol Architecture Reference Figure*, HEL.29, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S6] *AAL type 2 Addressing Considerations*, HEL.34, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S7] *AAL2 CID Contention Handling*, HEL.35, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S8] *Generic Primitives Provided for the AAL type 2 Signaling Protocol*, HEL.37, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S9] *AAL type 2 Signaling Message Flows for Basic Connection Handling*, HEL.39, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S10] *AAL type 2 Path Identifier*, HEL.40, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Helsinki, Finland, August 1998.

[S11] *Coding of SSCS Specific Information in Q.AAL2*, Delayed Contribution D.1103, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S12] *Blocking Procedure*, Delayed Contribution D.1105, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S13] *Basic Connection Control*, Delayed Contribution D.1109, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S14] *Proposed Modifications to Appendix 1 in Draft Q.AAL2*, Delayed Contribution D.1112, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S15] *Service Provided by the AAL Type 2 Signaling Entity*, Delayed Contribution D.1113, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S16] *Timers & Cause Codes in Q.AAL2*, Delayed Contribution D.1114, ITU-T Study Group 11 Working Party 1 Meeting, Ostend, Belgium, November 1998.

[S17] *Reuse of Cause Values Defined in Q.2610*, BER.20, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Bern, Switzerland, January 1999.

[S18] *Layer Management Interface Definition and Related Corrections*, BER.25, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Bern, Switzerland, January 1999.

[S19] *State Transition Diagrams for the Maintenance Protocol Procedures*, BER.26, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Bern, Switzerland, January 1999.

[S20] *A Correction in the Normal Connection Release Procedure*, BER.29, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Bern, Switzerland, January 1999.

[S21] *Draft New ITU-T Recommendation Q.sbcmtp AAL type 2 Signalling Bearer Converter on Broadband MTP*, BER.34, ITU-T Study Group 11 AAL2 Special Rapporteur's Meeting, Bern, Switzerland, January 1999.

[S22] *State Transition and SDL Diagrams for Signalling Bearer Converter on Broadband MTP*, Delayed Contribution D.1334, ITU-T Study Group 11 Meeting, Geneva, Switzerland, March 1999.

[S23] *Aligning the Cause Values Used in Q.AAL2 width Q.2610 Based on the Agreement Made in Bern*, Delayed Contribution D.1239, ITU-T Study Group 11 Meeting, Geneva, Switzerland, March 1999.

[S24] *Modifications to the GIT Recommendation to Support Identification of Q.AAL2 Users*, Delayed Contribution D.1240, ITU-T Study Group 11 Meeting, Geneva, Switzerland, March 1999.

[S25] *Delay Performance of AAL2 over a 2 Mb/s I_{ub} Link*, 3GPP TSG-RAN Working Group 3 meeting, Berlin, Germany, August 2000.

Independent Citations

[J2] is cited in the following publications:

[Cit1] S. Nananukul, Y. Guo, M. Holma, S. Kekki, *Some Issues in Performance and Design of the ATM/AAL2 Transport in the UTRAN*, IEEE Wireless Communications and Networking Conference, Chicago, USA, September 2000.

[Cit2] A-F. Canton, S. Tohme, D. Zeghlache, T. Chahed, *Performance Analysis of AAL2/ATM in UMTS Radio Access Network*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Lisboa, Portugal, September 2002.

[Cit3] A. B. Garcia, M. Alvarez-Campana, E. Vázquez, J. Berrocal, *Quality of Service Support in the UMTS Terrestrial Radio Access Network*, 9^{th} Workshop of the HP OpenView University Association, June 2002.

[Cit4] S. Dixit, Y. Guo, Z. Antoniou, *Resource Management and Quality of Service in Third-Generation Wireless Networks*, IEEE Communications Magazine, Vol. 39, No. 2, pp. 125-133, February 2001.

[Cit5] J. F. Huber, D. Weiler, H. Brand, *UMTS, the Mobile Multimedia Vision for IMT-2000: A Focus on Standardization*, IEEE Communications Magazine, Vol. 38, No. 9, pp. 129-136, September 2000.