Congestion Control in Integrated Services Networks

Masters Thesis

Loukas Rossides
Dept of Computer Science
University of Cyprus
1999

Contents

Chapters

Chapter 1

Introduction

1.1 Integrated Services Networks (ISNs)

From the beginning of networks we had a separate network deployment for each emerging service (e.g. telephone network, data network, cable TV network). Nowadays with the explosion of Internet and the need for real time services (voice, video, interactive multimedia applications) the need for an integrated services network has emerged. A network that will be capable for providing simultaneous transmission of different kind of services at different bitrates and with QoS, an Integrated Services Network (ISN). The first step towards this direction was ISDN (Integrated Services Digital Network, also called N-ISDN for narrow ISDN). ISDN provides up to 2 Mbits rate for a user so it can be used for basic video transmission but not for HDTV or other high bandwidth consuming applications. Soon it was obvious that ISDN was not offering what it promised and B-ISDN (Broadband ISDN) was proposed. The technology proposed as the basis for implementing B-ISDN was ATM (Asynchronous Transfer Mode). This is briefly described in section 1.1.1. Lately a lot of interest has also been generated toward transforming the Internet into an Integrated Services Network. These efforts are described in sections

1.1.2 and 1.1.3 In section 1.2 we offer a brief discussion on congestion control and in section 1.3 the current trends and future directions.

1.1.1 ATM

ATM stands for Asynchronous Transfer Mode. It is a highly efficient switching technique, able to offer connections for a wide range of different information type services at various bitrates. It is the underlying technology that makes B-ISDN a reality. The main features of ATM are:

- Connection oriented
- Small fixed-size packets called cells
- Offering of multiple services at various (and not fixed) bit rates

The cell size is set to 53 bytes from which 5 are used as a header and the rest are for user data. The selection of this size was a compromise between the 64-byte length cell proposed by USA forums and the 32-byte cell proposed by European forums. Small cell size was preferred in order to provide real time services. Real-time services require small delays and small losses acceptance. The 53-byte cell covers both requirements since it is small enough for fast switching (which means small delays) and loss of some cells means loss of a small and acceptable percentage of the total traffic sent. Offering support for real-time services is not something new in networks (recall POTS, ISDN). The novelty of ATM is that it can simultaneous support data, voice and video traffic with high transmission bit rates and with QoS guarantees. There is no need anymore for separate networks, e.g. telephone network, TV network, data network or quality compromise. Since ATM is one of the current ISN solutions, a general description of the way it manages traffic and congestion control follows in Section 2.4.

1.1.2 IP IntServ

Today's Internet is based on TCP/IP. The only service that TCP can currently offer is known as "best effort". As the need for new services has grown (voice, video, HDTV, teleconferencing etc) it is clear that the lack of service differentiation

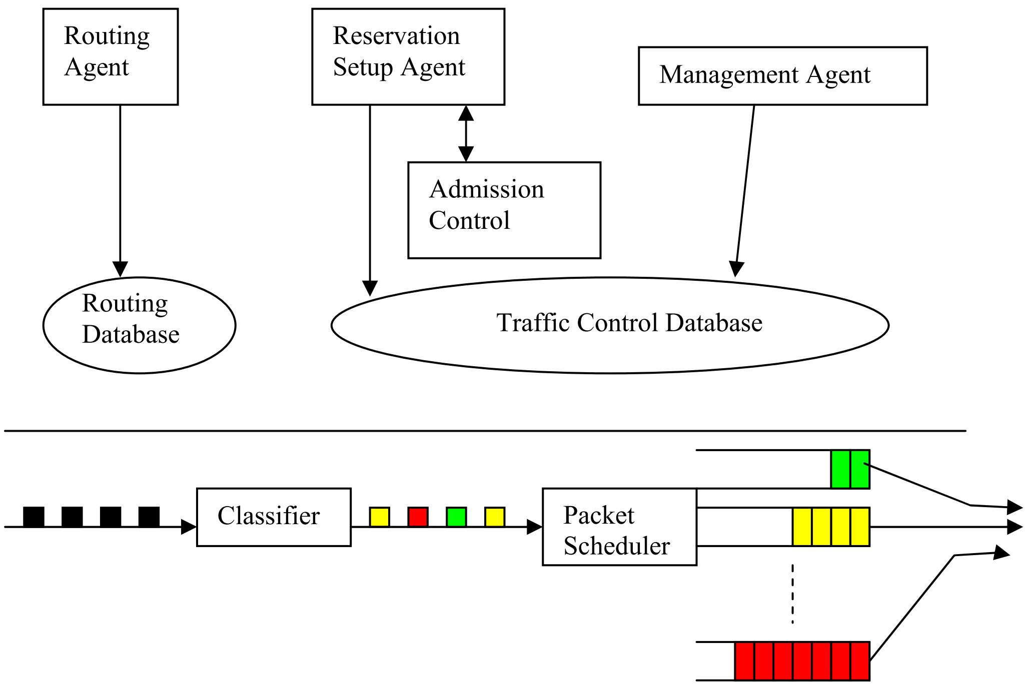

Figure 1.1: Integrated Services: Implementation Reference Model

in TCP accelerates the need for enhancements. As a result almost all high bandwidth consuming applications are moving from TCP and its incompetent congestion control mechanism to UDP, but as UDP offers no rate or congestion control mechanism the network becomes increasingly unprotected with unpredictable consequences to the network behaviour. For an excellent paper addressing this issue see [1]. The IETF (Internet Engineering Task Force) in an attempt to solve these problems has developed a number of enhancements to the current Internet infrastructure which allow the network to offer service differentiation. The first result of these efforts is the Resource Reservation Setup Protocol (RSVP) [2,3] and its associate suite of service classes [4, 5]. In this approach, individual applications signal their resource requirements to the network on an end-to-end basis. Given this the routers, switches etc in the network reserve the appropriate amount of resources needed by the application. Using the RSVP as the protocol for providing resource reservation IntServ [6] was proposed. IntServ stands for Integrated Services. An Integrated Services node consists of four parts:

- The packet scheduler
- The classifier
- Admission control
- Resource Reservation Setup Protocol

The packet scheduler is responsible for transmitting the different packets according to the resources that have been reserved for them. The classifier classifies the packets into different classes. All packets of the same class are treated equally by the scheduler and this classification is done at every node. Admission control has the responsibility of deciding based on the available resources whether or not to accept a new connection. Finally we have a protocol for signalling the resource reservation for the connection in the network. In IntServ case RSVP is used for this purpose. In Figure 1.1 we can see how the IntServ node works.

Another IntServ characteristic is the connection-oriented policy. With this we mean that for every session in the Internet firstly a connection must be established and then the user data transmission starts. This means that for every connection we made in the Internet we have to reserve resources in all the routers/switches that the connection traverses until it reaches its destination. As a result we may experience greater mean delays, waste of bandwidth and greater complexity. The reason is that most of the Internet connections are short in duration. Setting an end-to-end "reserved" connection by calling RSVP every time would probably consume more time than the actual connection duration itself. In other words for such a short-lived connection the overhead of setting it up is too high. Also there is the problem of old routers who might probably not support RSVP and the problem that maybe a router cannot afford the requested resources. This would mean extra delays for redefining

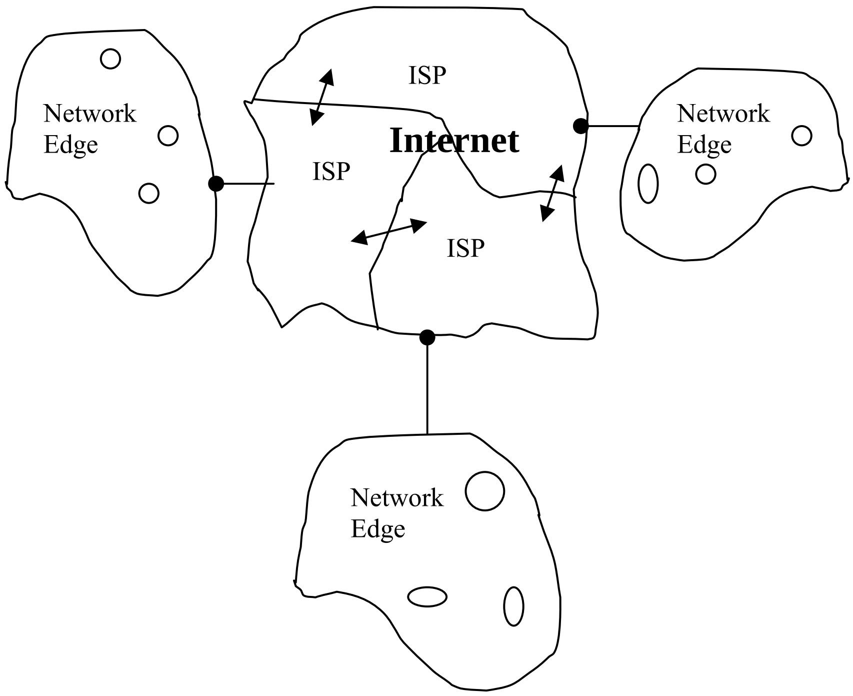

Figure 1.2: DiffServ Architecture

the source-destination path through routers that do have the necessary resources. Having these in mind is easy to understand why IntServ failed in establishing as an ISN solution for the Internet and why new solutions were proposed.

1.1.3 IP DiffServ

Since IntServ failed, the IETF proposed a more evolutionary approach that did not require significant changes to the Internet infrastructure and provided differentiation of services (Diff-Serv) [7]. To accomplish this Diff-Serv uses the ToS bits in the IP header, which are now renamed as "DS field" [8]. The functions associated with these bits have been also redefined. The main issue of the Diff-Serv approach is how to standardise a simple set of mechanisms for handling packets with different priorities set by the DS field in the IP header. In Figure 1.2 we can see the basic Diff-Serv architecture approach. Note that Diff-Serv works only at the edges of the network. This means that the priorities are set at the edges of the network, which reduces the complexity and makes it more scalable. On the other hand nothing is done to assure that the priorities will actually mean something when the packet enters the Internet (leaves the edge router). So we can say that Diff-Serv provides only a very basic QoS, without any quantified guaranties (as is the case in ATM). Because of the limited number of bits in the DS field, the Diff-Serv working group has defined a small set of building blocks, called per-hop behaviours (PHBs) which are used by the routers to deliver a number of services. They are encoded in the DS field and they specify the forward behaviour each packet will expect to receive by the individual routers in the Internet. When this will be used on an end-to-end basis it is envisioned that it will offer support to a number of emerging applications. The two PHBs being standardised are the Expedited Forwarding (EF) [9], and the Assured Forwarding [10]. The EF PHB specifies a forwarding behaviour in which packets see a very small amount of loss and a very low queuing delay. In order to ensure that every packet marked with EF receives this service, EF requires from every router to allocate enough forwarding resources so that the rate of incoming EF packets is always less than or equal to the rate at which the router can forward them. This is done through a Service Level Agreement (SLA) during the connection setup. In order to preserve this property on an end-to-end basis, EF requires traffic shaping and reshaping in the network. Although there is no specific method set for this, it will most probably done by a leaky bucket buffering algorithm. The AF PHB group specifies a forwarding behaviour in which packets see a very small amount of loss. The AF PHB group consists of four independently forwarded classes. Within each class, two or three drop preference levels are used to differentiate flows in the class. The idea behind AF is to preferentially drop best-effort packets and non contract conforming packets when there is congestion. By limiting the amount of AF traffic in the network and by managing the best-effort traffic appropriately, routers can then ensure low loss behaviour to packets marked with the EF PHB. As we can see Diff-Serv will try to provide some QoS using a drop-preference algorithm when congestion occurs. The most popular algorithm used for this purpose is RED and will be described later in section 2.5.1

1.2 Congestion and the need to control it

Ideally we want unlimited bandwidth for every user connected to the network. This is the dream not only of the users but also of network engineers. Reality however is different. We have limitations to the available bandwidth. Limitations that are not only because of the transport protocol but also of the medium for transferring informations (copper wires, electrical transmitters etc). Since the resources are limited and users and their demands are exponentially increasing we need to control the way network resources are used. We want to send as much traffic as possible through a link to maximise our throughput. On the other hand if we send at excessive rates then we will experience congestion. The network throughput will dramatically drop (Figure 1.3) and we may even have a collapse. To avoid such situation congestion control is needed. As it is shown in the following chapters current solutions fail to provide satisfactory control. The need for new technology and network independent, congestion control algorithms, is more demanding than ever. To satisfy these needs we must consider a new totally different approach based not necessarily on classical queuing theory. Dynamic models based on fluid flow models and non-linear control theory is a possible approach. A solution following this approach can offer mathematical correctness and protocol independence.

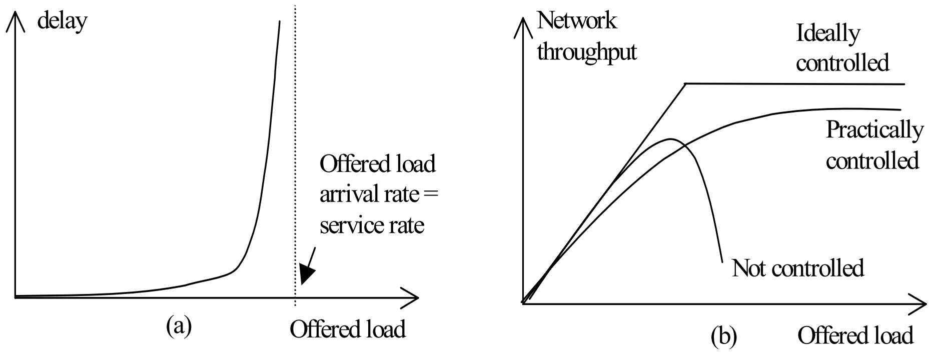

Figure 1.3 Network delay and throughput versus offered load

1.3 Current trends and future directions

As noted above current ISN solutions are ISDN, ATM and classical TCP/IP networks. Although TCP/IP does not strictly comply with ISN definition we must consider it as a solution since Internet and the majority of current networks use TCP/IP as their transport protocol. In its current form TCP/IP can support voice (voice over IP) and simple data transfer but without offering QoS. Extensive research is done for supporting multiple services in order to be classified as a "true" Integrated Services Network.

As the user needs for more bandwidth and QoS are becoming essential, networks must be upgraded not only in the transport protocol level but also in the physical level. Several are offered. They cover a wide area, from an all-optical network to hybrid systems with fiber optics as the backbone and copper wires or wireless termination to the user terminal. Recent advances in WDM (Wavelength Division Multiplexing) and photonics promise a full photonic network in the near future. In that case the limitations in bandwidth will be minimised and congestion control will be much easier if not unnecessary.

However, independent of the selected solution it is certain that in the next few years users will demand higher bandwidth connections (order of megabits) and guaranteed QoS. Photonic networks along with ATM seem to provide that but we have to consider the already established throughout the world IP network. Which solution or combination of solutions will be preferred it is difficult to predict. For the time being it is certain that congestion control is essential and crucial for implementing ISNs. Current solutions are not satisfactory and new congestion control schemes are necessary. Novel schemes that may not necessarily be based on queuing theory models and ad-hock approaches. The contribution of this thesis is such a novel approach for congestion control based on a dynamic fluid flow model and non-linear control theory. This, I consider, has an essential role to play in the future of congestion control for ISNs.

Chapter 2

Congestion control in ISN

2.1 General principles of congestion control

To develop congestion control, computer networks, like any other complex system, can also be modelled from a control theory perspective. These models attempt to capture the relationship between a cause and effect (input-output). In control system designs the control input is often manipulated to meet prescribed system performance. Based on a model, two approaches exist in designing controls: open and closed loop. Open loop solutions attempt to solve the problem by trusting the model accuracy. Such an accurate model that is tractable and allows the derivation of a control behaviour is extremely difficult and costly to develop. Current models proposed in the literature fail to adequately describe the behaviour of the system, in all its aspects. Thus making open loop based solutions very difficult if not impossible. Closed loop approaches use the concept of feedback to overcome this problem. Since feedback will attempt to correct any deviations we don't need an accurate model. This makes modelling easier than in open loop approaches. A proper closed loop solution must follow the next steps:

1. Monitor the system to detect when and where congestion occurs
2. Pass this information to the controller
3. Adjust system operation to correct the problem.

Although we have feedback and an accurate model is not needed (only one that captures the ‘dominant’ behaviour of the system), careful design of the controller is necessary otherwise instability may occur. The benefits and drawbacks of closed-loop solutions can be found in any classical control theory textbook. Feedback based control is in widespread use in systems ranging from a simple thermostat based temperature controller to nuclear plant controls or even the space shuttle.

2.2 Traditional queuing theory models

For the evaluation of performance, the description of the behaviour of queuing systems using mathematics (queuing theory) has been very successful. In traditional queuing theory we have the notation A/B/n where A represents the arrival rate distribution, B the service-time distribution and n the number of servers. Following this notation we can describe a number of queuing systems, such as M/M/1, M/G/1, M/M/n, M/D/1 etc where M stands for Markov (Poison arrivals, exponential service-time distribution), D for discrete and G for General distribution. This methodology was extensively used to analyse and model early networks (e.g. telephone networks) as well as data networks with acceptable results. The use of this methodology was always based on the assumption of static (non-dynamic) models. This however is not the case for modern networks where we have different demands on the network behaviour. Lately it has been demonstrated that data networks exhibit "chaotic" user and network behaviour. Demands placed on the network by real-time services exacerbate this situation further. This diverse traffic mix imposes differing on the end-to-end quality of service guarantees. Static models fail to describe this behaviour. Based on these reasons we seek an alternative modelling approach for congestion control.

2.3 Alternative models

Considering the above we propose that the new models need not be the same as the performance models traditionally derived from queuing theory. Performance models require accuracy, could be static and could be solved offline. However, for network congestion controls we require dynamic models, which must be solvable for designing controllers with good control properties. Current network models based on queuing theory simply do not offer this. The network is too complex to be accurately modelled, its states change dynamic and the model structure even changes, depending on the current network traffic mix. That's why we must consider dynamic models as for example fluid flow models as a possible solution. Such models need only capture the "essential" dynamic behaviour and be computationally easy for control system design. Such a model is presented in Chapter 3.

2.4 ATM Traffic Management and Congestion Control

Providing these capabilities with QoS means that ATM must have good congestion control and since ATM covers a broad range of services and QoS guaranties congestion control becomes more complex. To make things easier ATM Traffic Management and Congestion Control was separated into three categories:

- **ATM traffic contract**
- **Traffic control and ABR control**
- **Congestion indication and control.**

2.4.1 ATM Traffic Contract

With traffic contract we mean that an agreement between the user and the network for each connection is established before the user can send any data. This agreement covers several aspects of a cell flow such as a set of QoS parameters (Cell Delay Variation, Cell Loss Ratio, max Cell Transfer Delay etc), traffic parameters (Peak Cell Rate, sustainable cell rate, etc) and a conformance checking rule (leaky bucket based). This rule is needed in order to check if the connection is compliant with the agreement. The compliant connection definition is set from the network administrator before a connection is negotiated (sets buffer thresholds for CLR and leaky bucket). This is needed in order to check if there is any non-compliant connection. In such a case the network cannot offer any guaranties for QoS.

2.4.2 Traffic Control and ABR control

To monitor whether a connection is compliant or not, the definition of traffic control is introduced. We need traffic control not only to ensure that the users send traffic within the rates specified in their traffic contract but also to ensure that the network delivers to the users the negotiated QoS. To do that ATM uses a UPC policy based on leaky bucket and Generic Cell Rate Algorithm (GCRA), a GFC algorithm, Connection Admission Control (CAC) [11] and traffic shaping techniques. These algorithms are used in the case of CBR and VBR traffic to check the conformance of a connection. In the case of ABR we have a dynamic GCRA algorithm for connection conformance and a closed-loop flow control that uses the ER field in RM cells to instruct users at what rates to send in order to stay within the traffic contract and avoid congestion. ATM Forum has not defined what algorithm must be used in such cases. It only specifies a general framework for a rate-based closed-loop flow control mechanism [11]. Within this framework several feedback based control schemes have been proposed including BECN [12], EPRCA [13], CAPC [11], ERICA [14], FERM[43], ACC[41]. Most of them were based on results obtained from extensive simulations and intuition and we could say that they are ad-hoc proposals. This makes the analysis of such algorithms and their closed-loop behaviour very difficult if not impossible. Also the interaction of additional non-linear feedback loops can produce unexpected and erratic behaviour [15, 16]. Another drawback of these schemes is that they do not take into consideration the existence of real time traffic and do not see the control problem from a more general view but instead they focus on the ABR traffic and buffer metrics.

2.4.3 Congestion Indication and control

Apart from having traffic contract and traffic control we can have a more generic scheme indicating what actions should be made according to predefined network states. This scheme is Congestion Indication and Control. With this we mean that we can divide congestion status into three categories and states (no congestion, mild congestion, severe congestion) according to which we can apply one or more solutions for each ATM level [17] (see Table 2.1).

Category	Cell Level	Burst Level	Call Level
Management	UPC discard	Resource Allocation	Network Engineering, CAC
Avoidance	EFCI, UPC Tagging	Window, Rate or Credit Flow Control	Oversubscribed CAC, Call blocking
Recovery	Selective Cell Discard, Dynamic UPC	Loss Feedback, EPD/PPD	Call disconnection, Operations Procedures

Table 2.1 Congestion control, Categories and Levels

As we can see from Table 2.1, for each category we have different solutions. Each solution applies to a different ATM level. This table also, shows the complexity of congestion control. Management solutions are used in case of no-congestion within the network, avoidance solutions in case of mild congestion and recovery solutions in case of severe congestion (see Figure 2.1).

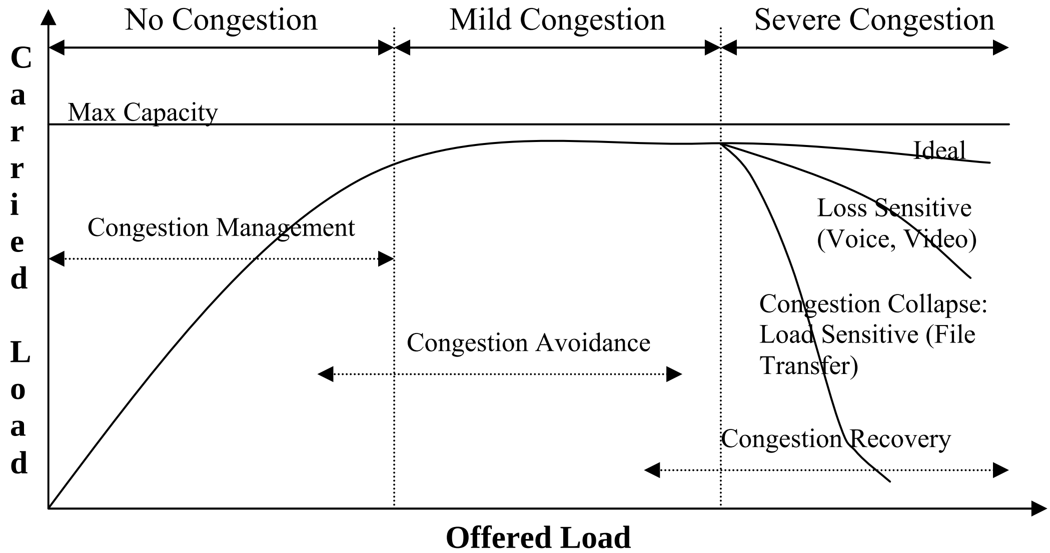

Figure 2.1 Illustration of congestion Regions and Collapse

Since our objective is to maximise throughput and minimise losses, we must concentrate our efforts in the mild congestion region and in congestion avoidance solutions. These solutions can be based on the following methods:

- Explicit Congestion Notification (ECN)
- Usage Parameter Control (UPC)
- Connection Admission Control (CAC)
- SVC Call Blocking
- Flow control

The latter method is the most important and will be described in more detail. Under this category we have three approaches. The window-based flow control, the rate-based flow control and the credit based flow control. All approaches must use methods that conform to the rules as defined in the ATM Forum Traffic Management v.4.0 specifications [18].

Window-based flow control is very simple. It was the first type of flow control used in data networks and TCP. Each source has a dynamically changing transmit window that determines how many cells a source can send during successive RTTs. While no cells are lost in the path, the source increases its window. Once a switch buffer overflows (actually surpass a buffer threshold), the sources decrease their transmission rate in a multiplicative way. As a result most window flow-controlled protocols exhibit the effect of increasing throughput and buffer utilisation followed by a period of rapid backoff.

In Rate-based flow control the feedback control loop controls the transmit rate of the sources instead of the window size [19]. Usually we select the transmit rate to start either from zero or from the maximum allowable rate and adjust it accordingly to the network state and feedback received. In its basic form we use (like in the window-based case) a buffer threshold or buffer length to provide feedback. However many simulations have shown that if we also use the rate of buffer growth to compare with the buffer length, we can get better results. For example, in its basic form the switch measures the fill rate of the buffer and every RTT (or other user defined period) compute a feedback to the sources to adjust their sending rate, according to a control strategy

Usually these kinds of algorithms provide better results in terms of throughput and delay variation than the window-based. This is why most known congestion control algorithms for ABR (EPRCA, ERICA, FERM) are under the rate-based category.

Finally we have the credit-based flow control [20]. In this method of control the parameter adjusted by the switch is the credit available for transmitting cells. A source may continue sending cells (each time they send a cell they decrease by one their credit counter) as long as the credit counter is greater than zero. Every RTT the switch sends a feedback message to each source indicating their credit. So a source can send cells (as many as it is entitled to do) every time (RTT period) it receives its credit. Because credit-based flow control dedicates buffers to each connection, other connections don't experience any impact from congestion. The advantages of this method are that it achieves almost 100 percent throughput and that it keeps the buffer relatively full. The disadvantage is that we have dedicated buffers for each connection (which means increased complexity at the switch), complexity at the sources for implementing the credit control logic and relatively large amount of buffer storage if we have longer than LAN propagation delays.

2.5 IP Congestion Control

As we have seen in the previous sections the existing congestion control solutions deployed in the Internet Transport Control Protocol (TCP) [21,22] are increasingly becoming ineffective, and it is generally accepted that these solutions cannot easily scale up even with various proposed "fixes" [23,24,25]. Also, it is worth pointing out that the User Datagram Protocol (UDP), the other transport service offered by IP Internet, offers no congestion control. However more and more demanding users use this for the delivery of real time video and voice services. The newly developed (also largely ad-hock) strategies [26,27,28] are also not proven to be robust and effective. Asynchronous Transfer Mode (ATM) has also witnessed a similar approach, with various congestion control schemes proposed [29,30]. Since these schemes are designed with significant non-linearities (e.g. two-phase—slow start and congestion avoidance—dynamic windows, binary feedback, additive-increase multiplicative-decrease flow control etc) based mostly on intuition, the analysis of their closed loop behaviour is difficult if at all possible, even for single control loop networks. Even worse the interaction of additional non-linear feedback loops can produce unexpected and erratic behaviour [31]. Empirical evidence demonstrates the poor performance and cyclic behaviour of the controlled TCP/IP Internet [32] (also confirmed analytically [33]). This is exacerbated as the link speed increases to satisfy demand (hence the bandwidth-delay product, and thus feedback delay, increases), and also as the demand on the network for better quality of service increases. Note that for WAN networks a multifractal behaviour has been observed [34], and it is suggested that this behaviour —cascade effect—may be related to existing network controls [35]. Based on all these facts it is becoming clear that new approaches for congestion control must be investigated. Approaches that can combine better performance and behaviour as well as formal behaviour analysis.

2.5.1 RED

As we saw in Section 1.2.3 Diff-Serv tries to provide QoS (by avoiding congestion) using a drop-preference algorithm. The most popular algorithm used for this purpose is RED (Random Early Discard) [36]. RED simply sets some min and max dropping thresholds for each class. In case the buffer queue size exceeds the min threshold,

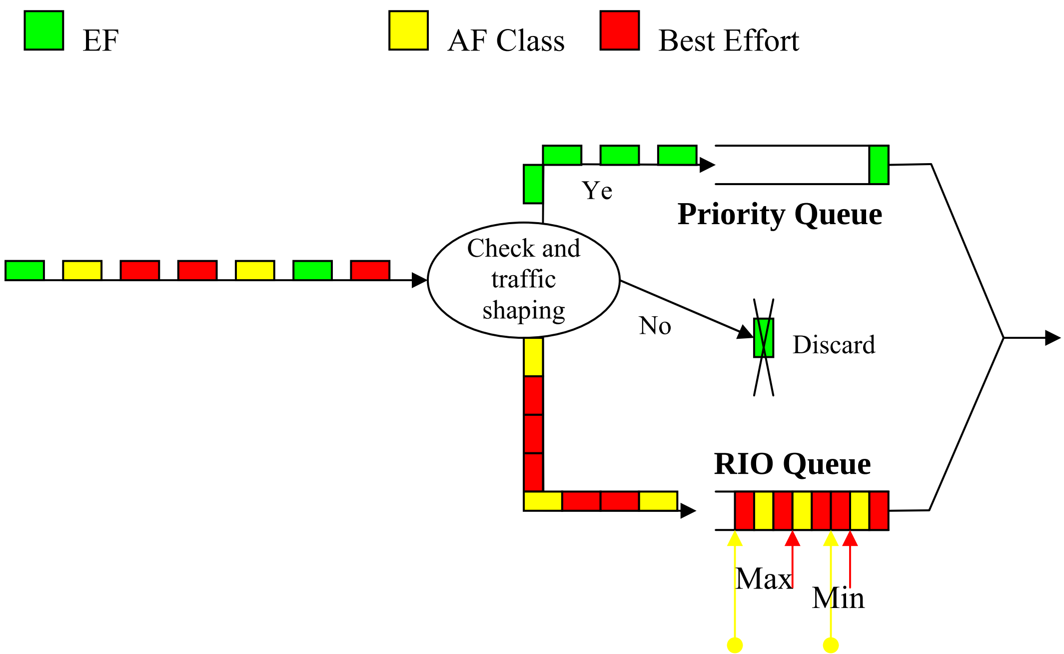

Figure 2.2: Diff-Serv scenario with RED queue for control

RED starts dropping randomly packets. If the buffer queue size exceeds the max threshold then every packet is dropped. The RED implementation for Diff-Serv defines that we have different thresholds for each class. Best effort packets have the lowest min and max threshold and therefor they are dropped with greater probability than packets of AF or EF class. Also there is the option that if an AF class packet does not comply with the rate specified it will be remapped to best-effort class packet. Apart from RED many other mechanisms such as n-RED, adaptive RED, BLUE [37] and Three colour marking were proposed for Diff-Serv queue control. In Figure 2.2 we can see a simple Diff-Serv scenario where RED is used for queue control. A leaky bucket traffic shaper is used to check if the packets comply with the SLA. If EF packets do not comply with the SLA then they are dropped. For AF class packets, if they do not comply then they are remapped into Best Effort Class packets. Both AF and Best effort packets are sharing a RIO Queue. RIO stands for RED In/Out queue, where In and Out means packets are In or Out of connection conformance agreement. For AF and Best Effort class we have different min and max thresholds. EF packets are using a separate high priority FIFO queue.

Chapter 3

IDCC:A non linear congestion control algorithm

3.1 Description and objectives

This section will try to investigate and propose such congestion control schemes using formal techniques like fluid flow and non-linear control theory. Although fluid flow based modelling for congestion control has been extensively studied, the combination of it with non-linear control theory has not been extensively studied. One may attribute this to the complexity of the control problem, coupled with the lack of collaboration between teletraffic engineers and control systems theorists. Recently several attempts have been made to develop congestion controllers using optimal control theory [[38]]; linear control [31,[39],[40]]; predictive adaptive control [[41],[42]]; fuzzy and neural control [[43],[44],[45],[46],[47]]; and non-linear control [[48],[49],[50],[51],[52]]. Despite these efforts the design of congestion network controllers whose performance can be analytically established and demonstrated in practice is still a challenging unresolved problem. We argue that the richness of non-linear control theory developed during the recent years justifies its use now. The most important recent developments in non-linear control theory include feedback linearisation [[53]], passivity theory [[54]], control Lyapunov functions [[55],[56]], backstepping and tuning functions [[57]], neural and fuzzy control systems [[58],[59],[60]], and robust adaptive control for linear and non-linear systems [[61],[62],[63]].

Early attempts to use non-linear control theory for network congestion control include [48-52]. The derivation of the control strategy was based on a simplistic nonlinear dynamic model of a network queue, derived from fluid flow considerations and from matching the M/M/1 queue behavior at equilibrium. It can be argued that this model cannot accurately predict the behavior of a network system. Even so, a study to design congestion controllers using this simplistic model which captures the 'dominant' dynamics [61] of the network system, but neglects secondary effects and the noisy environment has started. This section will present this approach and models studied.

Our objective is to create a model that can effectively and fairly share the resources among different classes of services and provide guarantees to the according classes. We will present a simple, dynamic fluid flow model that is promising in its ability to capture the essential dynamics of the system. Based on this model we will design a simple non-linear congestion controller. The model created for this purpose will attempt to keep a queue buffer length close to a reference value without knowledge or measurement of the flow in for both available (best effort) and guaranteed sources. We argue that the results are promising to encourage a more thorough study. We will present the Integrated Dynamic Congestion Controller that uses the models developed in the previous cases, extends them and combines them in order to provide guaranteed delays (and thus QoS) for real-time services and maximum efficiency for the best effort services. In other words we aim to achieve close to 100% throughput without sacrifying QoS for real-time services. In Chapter 4 we present results obtained from these studies. Future directions will be presented in Chapter 5.

3.2 A novel design approach

3.2.1 Description

In this section the model used for modelling ABR traffic competing with guaranteed traffic for the finite server capacity in ATM based networks will be presented. This model will be the basis for the models presented in section 3.3. A dynamic model is sought, in a form suitable for a distributed control solution. The objective is to find a model which captures the 'essential' dynamic behaviour, but has low order complexity relative to detailed probabilistic models such as the Chapman-Kolmogorov equations for determining the time-dependent state probability distribution for a Markovian queue [64]. Using the approximate fluid flow modelling approach proposed by Agnew [65], various dynamic models have been used by a number of researchers [64,[66],[67],[68]] to model a wide range of queuing and contention systems. Note that the fluid flow modelling principle has been extensively studied in the literature going back almost two decades with the interest still present until today [65,[69],[70],[71]].

Using the flow conservation principle, for a single queue and assuming no losses, the rate of change of the average number of cells queued at the link buffer can be related to the rate of cell arrivals and departures by a differential equation of the form:

$$\dot{x}(t) = -f_{out}(t) + f_{in}(t) \qquad (1)$$

Where:

$x(t)$ - state of the queue, given by the ensemble average of the number of cells $N(t)$ in the system (i.e. queue + server) at time t, i.e. $x(t)=E\{N(t)\}$

$f_{out}(t)$ - ensemble average of cell flow out of the queue at time t

$f_{in}(t)$ - ensemble average of cell flow into the queue at time t

The fluid flow equation is quite general and can model a wide range of queuing and contention systems as shown in the literature [64,67,66,68,[72]].

Assuming that the queue storage capacity is unlimited and the customers arrive at the queue with rate $\lambda(t)$, then $f_{in}(t)$ is just the offered load rate $\lambda(t)$ since no packets are dropped. The flow out of the system, $f_{out}(t)$, can be related to the ensemble average utilisation of the link $\rho(t)$ by $f_{out}(t)=C(t)\rho(t)$, where $C(t)$ is defined as the capacity of queue server. We assume that $\rho(t)$ can be approximated by a function $G(x(t))$ which represents the ensemble average utilisation of the queue at time t as a function of the state variable. Thus, the dynamics of the single queue can be represented by a non-linear differential equation of the form:

$$\dot{x}(t) = -G(x(t))C(t) + \lambda(t), \qquad x(0) = x_o \tag{2}$$

Different approaches can be used to determine $G(x(t))$. A commonly used approach to determine $G(x)$ is to match the steady-state equilibrium point of (2) with that of an equivalent queuing theory model where the meaning of "equivalent" depends on the queuing discipline assumed. This method has been validated with simulation by a number of researchers, for different queuing models [64, 67, 66]. Other approaches, such as system identification techniques and neural networks, can also be used to identify the parameters of the fluid flow equation.

3.2.2 Advantages

Most of the current congestion control methods are based on intuition and ad hoc control techniques together with extensive simulations to demonstrate their performance. The problem with this approach is that very little is known why these methods work and very little explanation can be given when they fail. The use of dynamic models could provide a better understanding of how the network operates and can be used to develop control techniques whose properties can be established analytically even when such techniques are based on intuition and ad hoc guesses. For control design purposes the model does not need to be accurate. It is because of the inability of modelling the real world accurately that feedback was invented and control theory is widely used. A good feedback control design should be able to deal with considerable uncertainties and inaccuracies that are not accounted for in the model. Robust adaptive control techniques for example can be used to control dynamical systems whose parameters are completely unknown and high frequency dynamics and disturbances are completely neglected in the control design [57,61]. A plethora of similar control techniques [53,57,61] and tools developed during the last decade offer a strong potential for solving complex congestion control problems in computer networks.

3.3 Derivative models

3.3.1 Dynamic Fluid Flow model

In this section we present as an example the dynamic fluid flow for representing ABR traffic in an ATM network. We will use this model to illustrate the design approaches for congestion control in the next section. We illustrate the derivation of the state equation for an M/M/1 queue following [64]. Assuming that the link has a First-In-First-Out (FIFO) service discipline and a common (shared) buffer, the following standard assumptions are made: the packets arrive according to a Poisson process; packet transmission time is proportional to the packet length; and that the packets are exponentially distributed with mean length 1. Then, from the M/M/1 queuing formulas, for a constant arrival rate to the queue the average number in the system at steady state is $\lambda/(C-\lambda)$. Requiring that $x(t)=\lambda/(C-\lambda)$ when $\dot{x}=0$, the state model becomes

$$\dot{x}(t) = -\frac{x(t)}{1+x(t)}C(t) + \lambda(t), \qquad x(0) = x_o \qquad (3)$$

The validity of this model has been verified by a number of researchers, including [66,67].

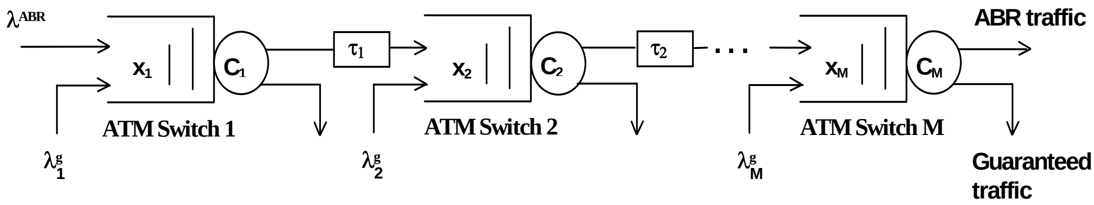

Figure 3.1. ABR traffic spanning M ATM switches competing with Guaranteed traffic for the common resources

We consider a series of *M,* ATM output buffered switching nodes that route cells from a set of incoming links to a set of outgoing links. The ABR traffic is modelled as a one-way connection between an Origin-Destination (OD) pair spanning M switching nodes (see Figure 3.1). The nodes are connected by links (1,...,*M*), associated with deterministic propagation delays τ_i (i = 1,...,*M*). Guaranteed traffic ($\lambda_1^g,..,\lambda_M^g$), appearing across every ATM switch, requires some access to the shared resources of each link. The queue at each link provides for statistical multiplexing of the incoming traffic streams.

Using fluid flow arguments, we can represent ABR traffic as a series of interconnected M/M/1 queues interfered by guaranteed traffic competing for the common resources. The ABR model, an extension of the M/M/1 queue model (1), is:

$$\dot{x}_1(t) = -C_1(t)\left(\frac{x_1(t)}{1+x_1(t)}\right) + \lambda^{ABR}(t) + \lambda_1^g(t)$$

$$\dot{x}_i(t) = -C_i(t)\left(\frac{x_i(t)}{1+x_i(t)}\right) + \gamma_i^{ABR}(t) + \lambda_i^g(t) \qquad i = 2,\dots,M. \tag{4}$$

where

$C_i(t)$ - bandwidth (capacity, cell service rate) allocated to the ABR traffic at node *i*,

$x_i(t)$ - state of the queue (i.e. ensemble average of number of cells in the queue) at node *i*,

$\lambda^{ABR}(t)$ - arrival rate due to ABR traffic

$\lambda_i^g(t)$ - arrival rate at cell-queue i due to Guaranteed traffic,

$\gamma_i^{ABR}(t)$ - ABR traffic entering node *i*, arriving from the previous node $i-1$, delayed by a deterministic amount τ_{i-1} due to the transmission propagation.

For $i = 2,\dots,M-1$

$$\gamma_i^{ABR}(t) = \left(C_{i-1}(t-\tau_{i-1})\frac{x_{i-1}^{ABR}(t-\tau_{i-1})}{1+x_{i-1}(t-\tau_{i-1})}\right).$$

where $x_i^{ABR}(t)$ is the ensemble average of the number of cell places in the buffer occupied by ABR traffic.

This model (4) can be used to represent all possible ABR traffic paths for any origin destination pair. The validity of using an M/M/1 queue to approximately describe the queuing delays with fixed packet length is discussed by Gerla et al [73]. Furthermore, in Chapter 4 we provide a simulative comparison and performance evaluation. Note that using similar arguments for packet based networks leads to the same fluid flow model [64].

3.3.2 Non-linear congestion controller

As a first step we have used the model (4) to derive and analyse a dynamic fluid flow model. In this way we verify the ability of this simplistic model to 'capture' the essential dynamics of the system, as well as the power of the non-linear control approach to meet the control objectives.

For the purposes of this model we consider a single ABR source, transmitting cells to a destination terminal. The cells traverse a number of ATM switches on route to the destination, interfered by guaranteed traffic competing for the common resources. We assume that buffer space and server capacity, up to a maximum (C_{max} and x_{max}) have been allocated to these services, which could represent a virtual path. Our objective is to control the buffer state to be close to a reference value, so as to indirectly guarantee the delay and loss for these services, an objective considered by many researchers [41,46]. We explore the possibility of dynamically controlling the server capacity. In this way, the source (when in need) is always guaranteed some resources, up to the allocated maximum (C_{max} and x_{max}), thus providing fairness to the source. However, whenever the source does not require the use of the (maximum) capacity and buffer space allocated (in order to maintain its QoS at the prescribed levels, due to very little or no traffic at this point in time) the controller allocates capacity dynamically, at a level that guarantees performance. The excess capacity, not currently used by the controlled source, can of course be used by other sources, thus allowing fair dynamic sharing of resources by the users. The control objective is to keep the buffer state $x_i(t)$ close to a reference value x_i^{ref}, and also ensure that the bounds on buffer space x_{max} and server capacity C_{max} are not exceeded. We assume that the input rate $\lambda^{ABR}(t)$ and $\lambda_i^g(t)$ are not known. The server capacity C(t) is the control variable. The dynamic allocation of server rate in an ATM switch shared by competing sources is investigated.

The selected control strategy for switch 1 is developed using the model (4) as follows [74]:

Let

$\bar{x}_1 = x_1 - x_1^{ref}$, then $\dot{\bar{x}}_1 = \dot{x}_1$

Then from (4)

$$\dot{\bar{x}}_1(t) = -C_1(t)\left(\frac{x_1(t)}{1+x_1(t)}\right) + \lambda_1^{ABR}(t) + \lambda_1^{guaranteed}(t) \tag{5}$$

Using feedback linearization and robust adaptive control ideas [53,57,61] we choose the control input i.e. capacity as

$$C_1 = \rho_1 \frac{1+x_1}{x_1}\left[\alpha_1 \bar{x}_1 + k_1\right]$$

where (6)

$$\rho_1 = \begin{cases} 0 & if \;\; x_1 \le 0.01 \\ (100/99)x_1 - (100/99 - 1) & 0.01 < x_1 \le 1 \\ 1 & x_1 > 1 \end{cases}$$

$$\dot{k}_1 = \Pr\left[\delta_1 \bar{x}_1\right], \qquad 0 \le k_1(0) \le \widetilde{k} \tag{7}$$

$$\Pr[\delta_1 \bar{x}_1] = \begin{cases} \delta_1 \bar{x}_1 & if \quad (0 \le k_1 \le \tilde{k}_1)\, or\ (k_1 = \tilde{k}_1\ and\ \bar{x}_1 \le 0)\ or\ (k_1 = 0\ and\ \bar{x}_1 \ge 0) \\ \\ 0 & else \end{cases}$$

$\alpha_1, \delta_1 > 0$ and $\tilde{k}_1 \ge \lambda_1^{ABR} + \lambda_1^{guaranteed}$ are design constants.

From (6) and (7) we have

$$\dot{\bar{x}}_1 = -\rho_1 \alpha_1 \bar{x}_1 - \rho_1 \hat{k}_1 + \lambda_1^{ABR} + \lambda_1^{guaranteed} \rho_1 k_1^*$$

where

$$k_1^* = \lambda_1^{ABR} + \lambda_1^{guaranteed}, \qquad \hat{k}_1 = k_1 + k_1^*, \ \text{ and } \ \dot{k}_1 = \Pr[\delta_1 \bar{x}_1].$$

Assuming k_1^* is constant (the case where k_1^* is not constant can also be handled but the analysis is more complicated) we propose the following Lyapunov function for analysis.

$$V = \frac{\bar{x}_1^2}{2} + \frac{\hat{k}_1^{\,2}}{2\delta_1}$$

The time derivative of V along the solution of (6), (8) is given by

$$\dot{V}_1 = -\rho_1 \alpha_1 \bar{x}_1^2 - \rho_1 \hat{k}_1 \bar{x}_1 + \bar{x}_1 (\lambda^{ABR} + \lambda_1^{guaranteed} - \rho_1 k_1^*) + \hat{k}_1 \Pr[\bar{x}_1]$$

$$= -\rho_1 \alpha_1 \bar{x}_1^2 + \bar{x}_1 (\lambda^{ABR} + \lambda_1^{guaranteed} - \rho_1 k_1^*) + \hat{k}_1 (\Pr[\bar{x}_1] - \rho_1 \bar{x}_1)$$

which may be written as

$$\dot{V} = \begin{cases} \bar{x}_1 (\lambda^{ABR} + \lambda_1^{guaranteed}) + \hat{k}_1 \Pr[\bar{x}_1] & if \quad 0 \le x_1 \le 1 \\ -(x_1 - 1)\alpha_1 \bar{x}_1^2 + \bar{x}_1 (\lambda^{ABR} + \lambda_1^{guaranteed} - (x_1 - 1) k_1^*) + \hat{k}_1 (\Pr[\bar{x}_1] - (x_1 - 1)\bar{x}_1) & if \quad 1 < x_1 \le 2 \\ -\alpha_1 \bar{x}_1^2 + \bar{x}_1 (\lambda^{ABR} + \lambda_1^{guaranteed} - k_1^*) + \hat{k}_1 (\Pr[\bar{x}_1] - \bar{x}_1) & if \quad \bar{x}_1 > 2 \end{cases}$$

Examining the properties of $\dot{V}$ we can show that:

$\dot{V} = 0$ for $\bar{x}_1 = 0$, and $\dot{V} < 0$ for $\bar{x}_1 \ne 0$.

Hence from Lyapunov theory and additional arguments we have that, $\bar{x}_1$ is bounded and $\bar{x}_1(t) \to 0$ as $t \to \infty$. With the above choice of C_1 we can establish that the maximum possible value $C_{1,\max}$ of C_1 satisfies

$$C_{1,\max} \leq 2\left[\alpha_1 (x_{1,buf} - x_1^{ref}) + \bar{k}_1\right]$$

where α_1 and $\bar{k}_1$ are design constants which provide some flexibility in meeting possible constraints.

Generalising for a path consisting of M switches we have $C_i = \rho_i \dfrac{1+x_i}{x_i}\left[\alpha_i \bar{x}_i + k_i\right]$

The analysis of C_i is the same as the analysis of C_1. Only the indices of the variables are replaced by i and $\lambda_1^{ABR} + \lambda_1^{g}$ is replaced by $\lambda_i^{g} + \gamma_i$ where i=2…M. The simulative performance evaluation is presented in section 4.2.

3.4 The algorithm

Based on the models developed in the previous cases and the encouraging performance results, an integrated dynamic congestion controller strategy is developed. It provides guaranteed delays (and thus QoS) for real-time services and maximum efficiency for the ABR and best effort services. It can be seen as a generic scheme for handling multiple differentiated classes of traffic, using an integrated dynamic congestion control approach, derived using non-linear control theory and the fluid flow model presented in section 3.3.1. By differentiating each class, the control objective for each class is decoupled from the rest, thus simplifying the control design for each one. Each class can have a different control objective. Three illustrative classes are described in detail below. The control strategy is model based dynamic feedback linearization scheme with proportional plus integral action. It should be noted that the methodology used is general and independent of technology, as for example TCP/IP or ATM. Generically, we will use the term packet for both IP packets and ATM cells, and switch for ATM switch and IP routers.

We divide traffic into three basic types of service (conceptually similar to the proposed DiffServ architecture for the Internet, i.e. Expedited Forwarding, Assured Forwarding and Best Effort): Guaranteed Service, Rate Controlled Service, and Best Effort Service. Each service transmits packets to destination terminals. The packets traverse a number of switches on route to the destination. At each switch we assume that dedicated buffer space is allocated for each one of the three services and that the server can be shared between the three in a controlled fashion (see Figure 3.2 below). Guaranteed Service requires strict guarantees of delivery, within given delay and loss bounds. It does not allow regulation of its rate (or at least regulation that will affect the given delay bounds). Any regulation of this type of traffic has to be achieved at the connection phase. Once admitted into the network the network has to offer service in accordance with the given guarantees. This is the task of the Guaranteed Traffic Controller. Rate Controlled Traffic on the other hand allows the network to regulate its flow (pace it) into the network. It cannot tolerate any losses of packets. It can however tolerate (variable) queuing delays. This is the task of the Rate Controlled Traffic Controller. Best Effort Service on the other hand offers no guarantees on either loss or delay. It makes use of any instantaneous leftover capacity.

For Guaranteed traffic, our approach is to tightly control the buffer state to be always close to a reference value, chosen by the network operator, so as to indirectly guaranteeing acceptable bounds for the maximum delay and loss. The capacity for the Guaranteed Traffic is dynamically allocated, up to the physical server limit or an agreed maximum. In this way, the Guaranteed Traffic is always given resources, up to the allocated maximum (C_{max} : the maximum available capacity and x_{max} : maximum buffer size) to ensure the provision of Guaranteed Service with known bounds. Due to the dynamic nature of the allocated capacity, whenever this service does not require the use of the maximum capacity in order to maintain its QoS at the prescribed levels it offers the excess capacity to the Rate Controlled Service.

The Rate Controlled Service regulates the flow of Rate Controlled Traffic into the network, by monitoring the queue state and comparing it to a reference value (could be chosen by the network operator). Then it uses a non-linear control strategy to inform the sources of the allowed rate they can transmit over the next control interval. The capacity for the Rate Controlled Service is dynamically allocated, up to the physical server limit minus the capacity given to the Guaranteed Service.

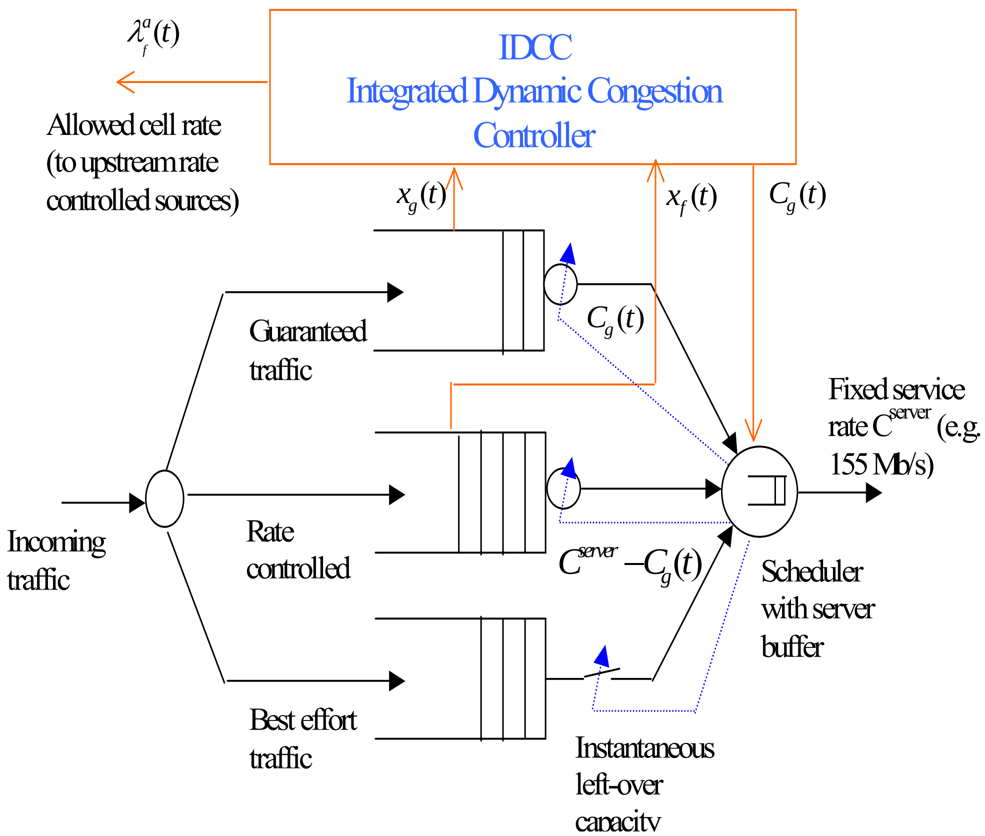

Figure 3.2 IDCC

The Best Effort service operates at the packet/cell scale and uses any instantaneous left over capacity. This is achieved by monitoring the combined server buffer at the server scheduler. In the absence of any packets in the queue awaiting transmission it allows a packet from the Best Effort Service to enter the server buffer (server buffer has a maximum of 2 packets; one in service and 1 in queue). Note for ATM this function is trivial, but for variable size packets (as in the Internet) more care is required so that real time packets are not caught behind very large Best Effort packets, for example by also monitoring the queue size of the other two services.

Chapter 4

Simulative Performance Evaluation

4.1 Simulation Program

The network simulator OPNET [75] is used to validate the models for a wide class of network interconnections and/or identify some of the unknown parameters and non-linearities. OPNET is a commercially available network simulation tool, commonly acknowledged as one of the leading solutions for modelling and simulation of communications networks, devices, and protocols. It features an extensive library of models for both ATM and TCP/IP, and its open architecture allows custom code to be integrated with the system. The main characteristics of OPNET are:

- Easy to use Graphical User Interface
- Powerful object-oriented simulation environment
- Data analysis integrated tools
- Hierarchical Modelling capability based on objects
- Option of using C like programming for defining own protocols, algorithms and models.

OPNET is an event-based simulation tool so it can provide us the capability for tracing step-by-step the execution of the model and compare the results with the behaviour expected through mathematical analysis. This makes easier debugging and correction of code. It provides also the possibility for capturing any parameter value of the model simulated, which makes analysis and presentation of results much easier. The fact that OPNET provides the majority of the current network equipment and protocols (routers, ATM switches, wires, optical fibers, protocol) based on vendor specification means that simulation are very close to reality. Also the user can define its own parameters such as propagation delay, delay variation, behaviour of sources, etc to make simulation scenarios as close to reality as possible.

4.2 Simulation Scenarios

4.2.1 Dynamic Fluid Flow model

In all scenarios an ATM topology is used. For the simple Dynamic Fluid Flow model a single ABR source is used along. There is no real-time traffic. The ABR traffic traverses through three ATM switches to reach the destination terminal. The configuration for this scenario is shown in Figure 4.1. This section provides a simulative comparison of the fluid flow model derived in section 3.3.1 with a discrete event cell based simulation (using OPNET) of an ATM switch fed by a bursty source (Fig. 4.1).

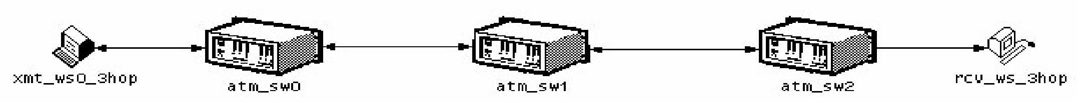

Figure 4.1 Network Topology of simulated model

We compare the time evolution of the state of the queue system, as given by the solution of the fluid flow model presented in section 3.3.1, with the state of the queue observed from the discrete event cell based simulation of an ATM switch.

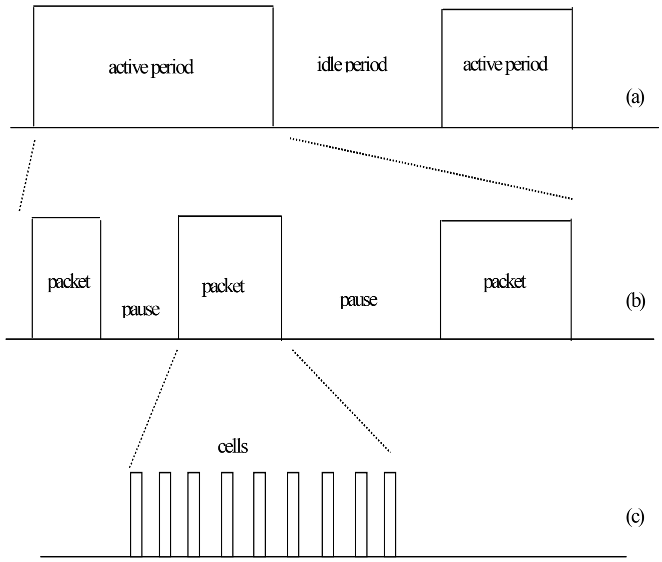

Figure 4.2 Traffic source model (a) Connection activity (b) Packet activity (c) Cell activity

Both model and simulation consider the same input representing a bursty on-off source, shown in Figure 4.2 (general) and Figure 4.3 (actual OPNET output). The active and idle periods of the connection are shown in part (a), the packet activity in part (b), and the cell activity in part (c) of Figure 4.2. The idle period has a geometric distribution with the mean value chosen to adjust the network load. During each active period a number of packets are generated with a geometric distribution and mean number of packets N. The packet size also has a geometric distribution with mean size of 8 Kbytes, while the pause period is exponential with mean value equal to 0.5 msec. For ATM based networks, each packet is segmented into cells and transmitted at a constant cell rate of 342170 cells/sec.

The time evolution of the queue state from both the model and OPNET simulation are presented in Figure 4.4. From Figure 4.4 we can observe that there is a reasonable agreement between the proposed model and the observed one, which demonstrates confidence to the model.

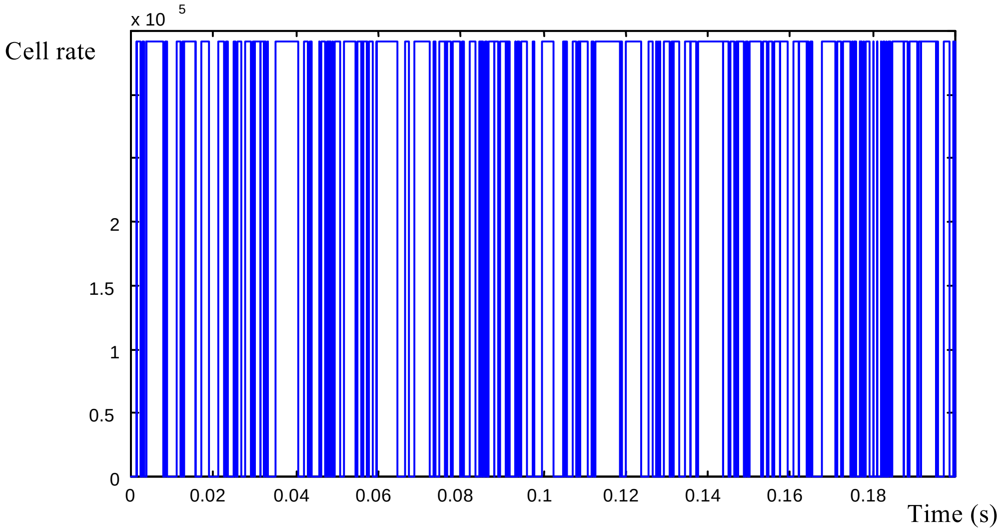

Figure 4.3 ABR source on-off periods at a cell rate of 342170 cell/sec

From Figure 4.4 and Figure 4.5 we can depict that the error between the MATLAB and OPNET simulations is between 0%-10% in most cases. We must take into account that the error presented is the difference between the actual queue size of the MATLAB and OPNET queue. So an error of zero indicates identical values between the two queues. We can observer that throughout the entire simulation sequence in only a small period (0.15-0.16 sec) we have a significance difference between the two queues (10%-60% error). This is clearly shown in Figure 4.5 where we can see that this difference is a result of an instantaneous burst around 0.16 sec.

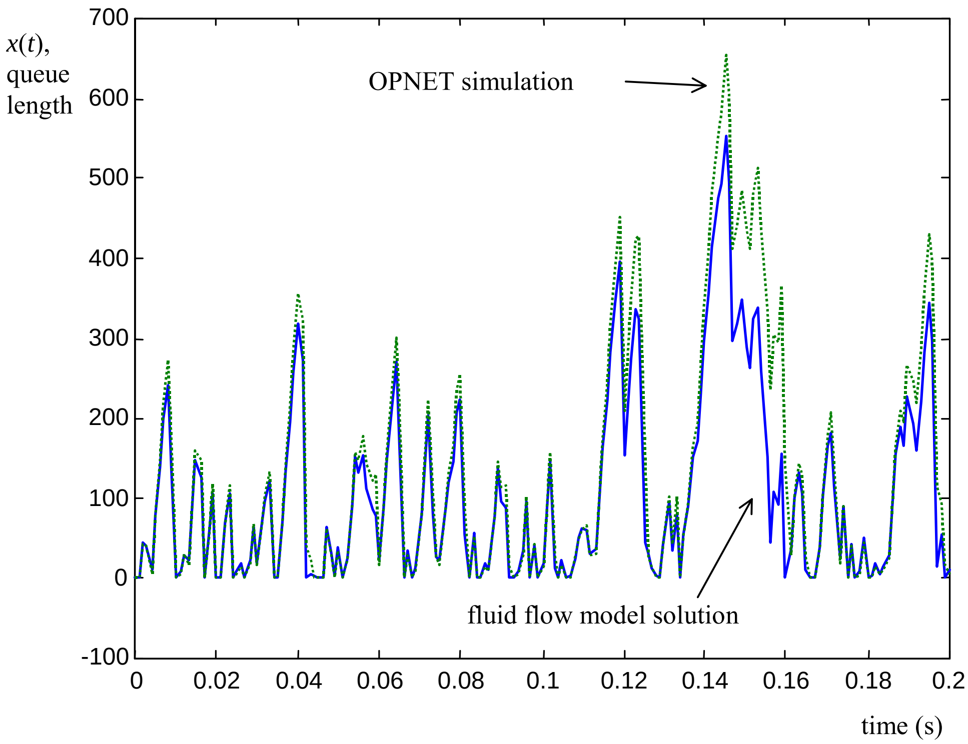

Figure 4.4 Time evolution of network system queue state obtained using OPNET simulation (broken line) and solution of the fluid flow model (solid line)

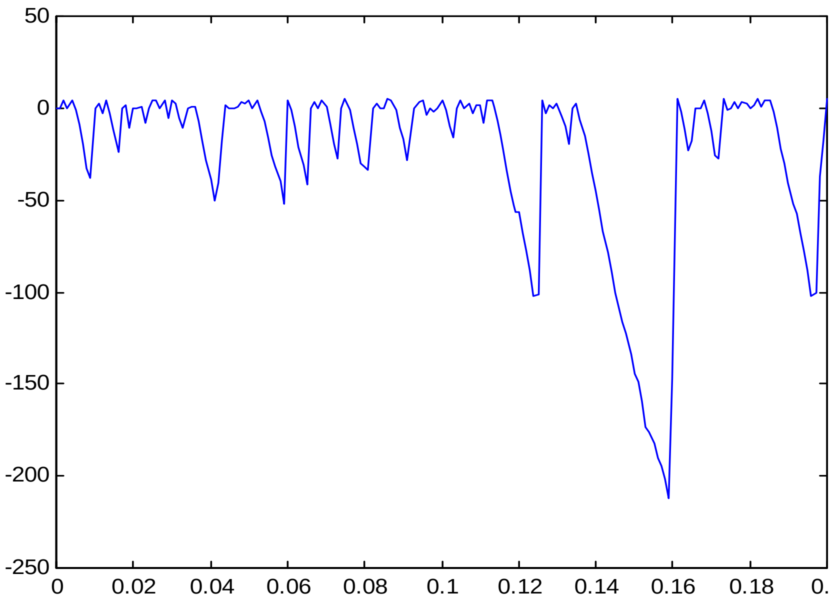

Figure 4.5 Model error (matlab model queue size–opnet model queue size)

Note that similar fluid flow models in both a discrete and continuous form have been used by a number of researchers for designing, or analysing the behaviour of network systems under control [[76],31,39,33,[77]]. For example, Rohrs [31] using similar fluid flow arguments derived the following discrete fluid flow model of the state of the buffer at the output port of an ATM switch.

$$x(nT) = \max\{x(n-1)T) + (\lambda(n-1)T - C(n))T, 0\} + n_q \tag{8}$$

where T is the sampling period, and n_q is a noise term representing the difference between the model and the actual queue system, and uses this model to evaluate the performance of a binary Backward Explicit Congestion Notification (BECN) control algorithm. He demonstrates the undesired cyclic behaviour of the controlled system. This (undesired) cyclic behaviour is also presented in [33] for TCP/IP, using dynamic models of the behaviour of the different phases of the TCP/IP congestion algorithms (slow start and congestion avoidance phase) for high bandwidth-delay products and random loss. Their results are demonstrated using simulations. In [77] for ATM congestion control they make use of a similar model, as given by (5), and using intuition they design an ABR flow control strategy (referred to as queue control function) to keep the queue well controlled, according to some known function (step, linear, hyperbolic and inverse hyperbolic). They use analysis and simulation to evaluate the proposed strategy. It is worth noting that many other types of models have been proposed, either using queuing theory arguments, or others, but in most cases the derived models are too complex for deriving simple to understand and implement controllers. As a result, motivated by the desire to derive simple controllers, the dynamic aspects of the network system are often ignored. For example, in [[78]] the analysis of the performance of simple (binary) reactive congestion control algorithms is carried out using a queuing theory approach model, which is limited to steady state analysis only due to the inability to handle the resultant computational complexity for the dynamic case.

4.2.2 Non-linear Congestion Controller

We evaluate the performance properties of the proposed strategy using a 3-node network model, shown in Figure 4.6. The ws0_3hop source is an On-Off source modeled as shown in Figure 4.3, with a peak cell rate of 110 Mbit/s. The mean pause period between packets is 0.0005 seconds and the mean number of IP packets per period is 10. The variable bit rate sources vbr_1hop_e, f, g, h are modeled by AR models representing video conferencing sessions, as suggested in [79], with a variable rate of between 2 to 14 Mbits/sec each. Each ATM switch has a finite output buffer of 1024 cell places with a reference point set to 400 cell places. The controller design variable α is selected equal to 10000, and the ISS Filter period (for calculating new Service Rate) is set at 32 cell times.

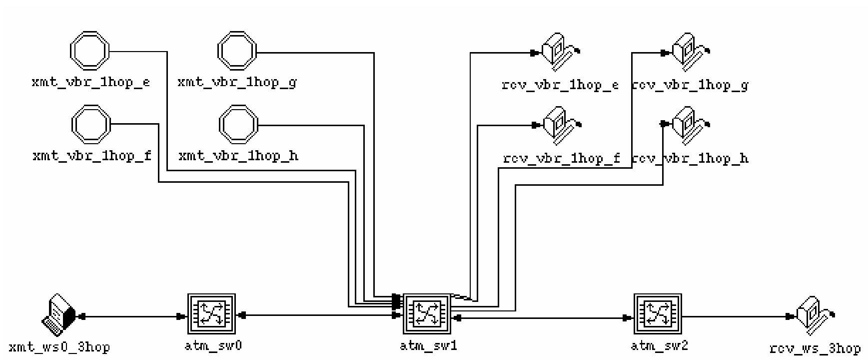

Figure 4.6 Three node network topology used for OPNET simulation.

The results obtained using OPNET simulations of the queue size, and the dynamically allocated (calculated) service rate (see Figure 4.7 and Figure 4.8) show that the network system is well controlled. The queue sizes remain at a constant level of 400 cell places, as dictated by the reference value. The simulation of the simple model (4) together with the controller (7) using MATLAB compare very well with the OPNET simulation demonstrating the validity of the model and robustness of the proposed controller with respect to model uncertainties and noise effects.

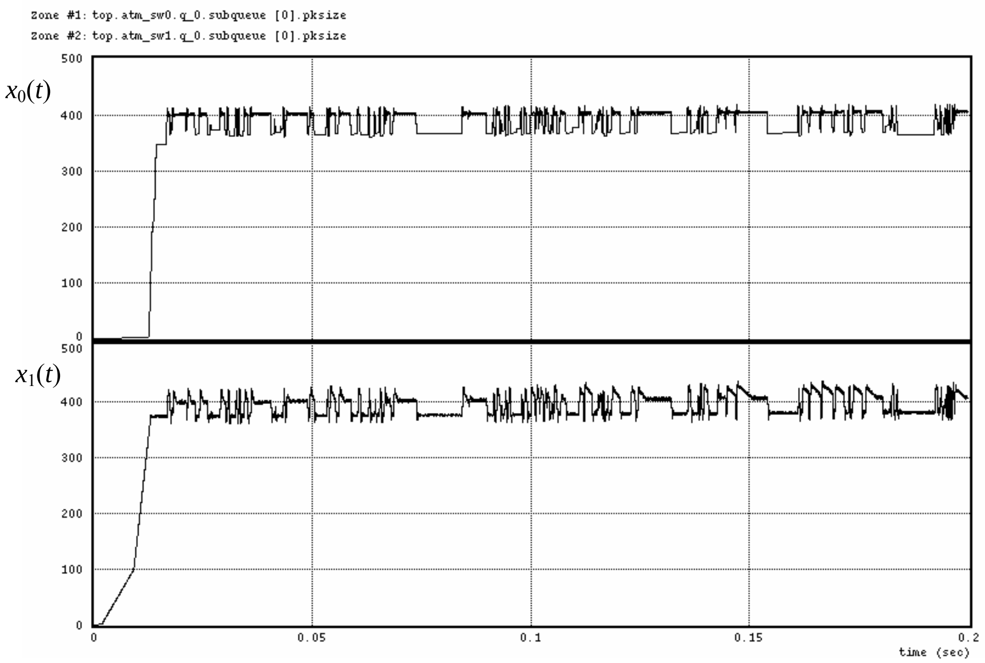

Figure 4.7 Buffer state (queue length in cells) for atm_sw0 and atm_sw1, using OPNET simulation: Reference point is set at 400 cell places.

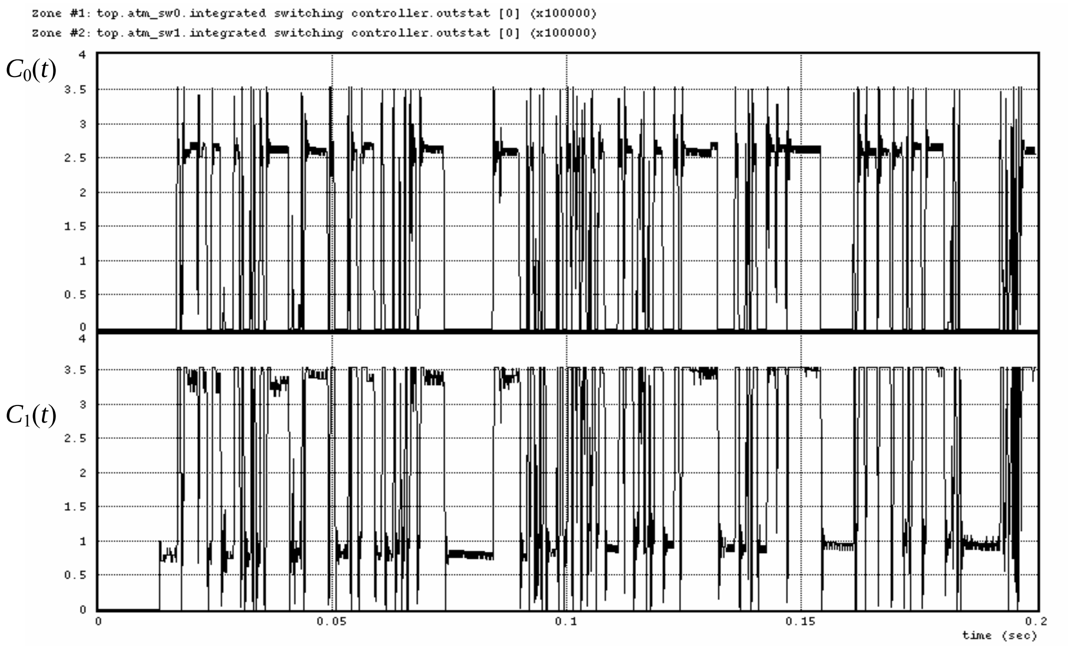

Figure 4.8 Dynamically allocated service rate for atm_sw0 and atm_sw1; OPNET simulation.

The above preliminary control design for capacity allocation is an initial successful step in dealing with the overall congestion control problem. Despite the simplicity of the model, the controller was able to deal with the uncertainty of the model and incoming traffic and meet the control objective. The presence of design parameters in the controller provides flexibility for further improvement in performance by tuning these parameters. The objective for future work is to exploit these preliminary results in an effort to come up with a systematic approach for designing congestion controllers for TCP/IP and ATM networks. The following section presents a proposed controller strategy based on the model presented in section 3.4

4.2.3 Integrated Dynamic Congestion Controller

This section presents the Integrated Dynamic Congestion controller (IDCC) model, which is based on the model presented in the previous sections. We have seen that the basic model (equation 4) captures the "essential" dynamics (see section 4.2), needed for creating a simple non-linear congestion controller (see section 3.4). There we saw that it can control the cell service rate in such a way so that we have a close to the reference point queue length for our ABR buffers. This means that we can have controlled cell delays in a buffer, which is what real-time services want. Having these in mind the IDCC based strategy was developed. Here we again use a three ATM switch path (which can be replaced by three IP routers; see later discussion). At each switch we have 10 ABR sources except the first switch where we have 20 ABR sources (10 are used as 1-hop and the other 10 as 3-hop). All other ABR sources are used as 1-hop sources. At the last switch we have an additional 4 VBR sources and 2 CBR sources. We also added a UBR traffic generator at the queues of each switch so that we have UBR cells in the switches available when there is not enough traffic to send. This way we have a combination of real-time and best effort traffic. This scenario is similar to an IP Diff-Serv case where we have the EF PHB (CBR/VBR in our case), AF PHB (ABR in our case) and Best-effort (UBR in our case). So we can claim that this model could also be ported for IP network or whatever network can operate in a similar way. In Figure 4.9 we can see the network topology as modelled using OPNET.

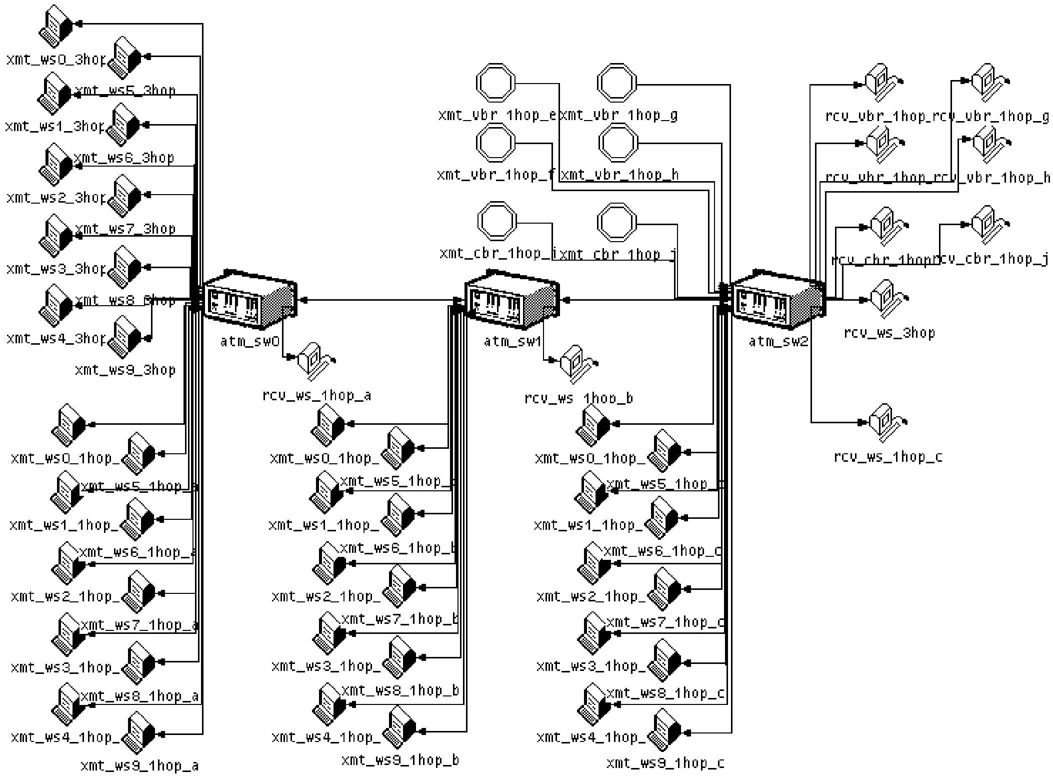

Figure 4.9 IDCC scenario, network topology

In this scenario CBR/VBR sources have a priority. We can guarantee them maximum delays not exceeding in any case the set by network administrator reference point. In order to test the quick responsiveness of our controller we set a variable reference point for this service. At the beginning we set the reference point to 100 cells. After t=0.4 sec it is set to 50 cells and after t=0.8 sec it is again raised to 100 cells (where t stands for time in seconds). This way not only we show that our controller can match the reference values but that it can also cope with dynamic changes that occur in the network (e.g. another connection is set-up, more bandwidth is required for real-time services etc). To make things more realistic we did the same for the ABR queues. So the ABR queues also have variable in time buffer reference point. For the ABR case where we can accept bigger delays we set the reference point to be 900 cells while t<0.3 sec. After that time it is set to 300 cells until t=0.7

sec and after t=0.7 sec it is raised to 600 cells. Note that the changes in the reference value occur out of phase with the changes in he reference value of the CBR/VBR. We test a variety of cases in order to allow as to evaluate the responsiveness and robustness of our control design. For example at t=0.30 sec the ABR queue is 900 cells. At t=0.31 we want ABR queue to drop down to 300 cells. This means that we can either increase the switch service rate available for ABR traffic or regulate at the ABR sources to send less traffic. Since at t=0.31 sec the CBR/VBR reference point is unchanged this means that we don't have spare resources for ABR thus we have to limit the sending rate. Things get more complicated at t=0.4 where real-time services get more demanding (the reference point is dropped from 100 cells to 50 cells). This means that the ABR sources must again drop their sending rate since extra capacity is allocated to serve Guaranteed Traffic. At t=0.7 ABR sources have more bandwidth available since the reference point again changes and is now raised at 600 cells. Considering the above and from the fact that the Guaranteed Traffic sources send traffic in an unspecified way the results shown in Figures 4.10, 4.11, 4.12, 4.13 demonstrate the responsiveness and robustness of the proposed congestion control strategy. In Figure 4.10 we can see the queue length in the ABR and real-time services buffers. It shows that the controller adapts very quickly to the set reference point.

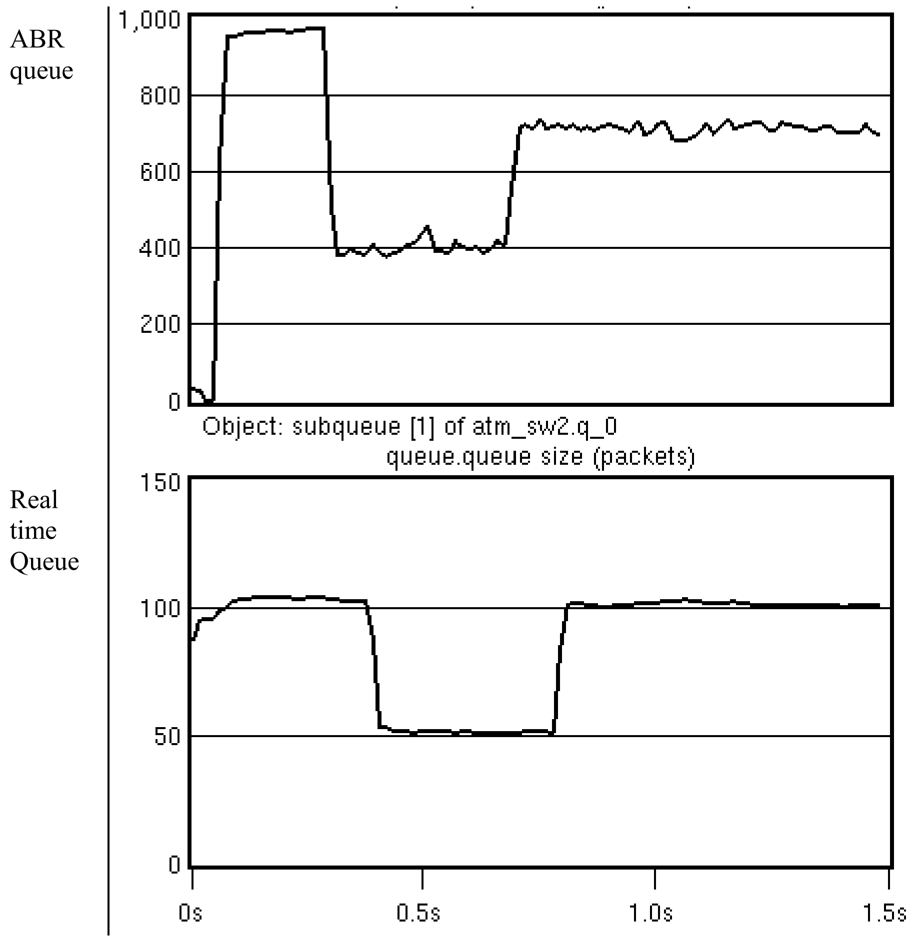

Figure 4.10 Switch 2 (last switch) Queues

For the case of real-time queue it matches exactly the reference point (100 cells and 50 cells). Observe that in the case of ABR is has a 100 cell offset for each case (900, 300, 600 cells reference point) which is no problem since it is a steady behaviour showed at all cases and the addition of integrating action in the controller design can rectify this. It is very important to notice that there are not transient over or undershoots, no oscillations or variance in behaviour and that it responds very quickly to the changes set in both queues. So we can say that we can dynamically control the buffer state and the sources sending rate which means that we can control the network effectively in a known fashion (as set by references) and therefor control congestion. In Figures 4.11 and 4.12 we see the queues for switch 0 and switch 1 (first and middle). The reference point for these switches is set to a constant 600 cell (since there is no real-time traffic there is no point to have a variable in time reference point). We can see that again we have an offset in the buffer queue length but same argument as earlier can be motivated here. The important observation is that while we have 3-hop sources that traverse across all switches with some considerable propagation delay the behaviour is well controlled. Note that the 3-hop sources sending rate is calculated based not only on the first switch status but mainly on the last switch where we have real-time services that use most of the available bandwidth (bottleneck switch).

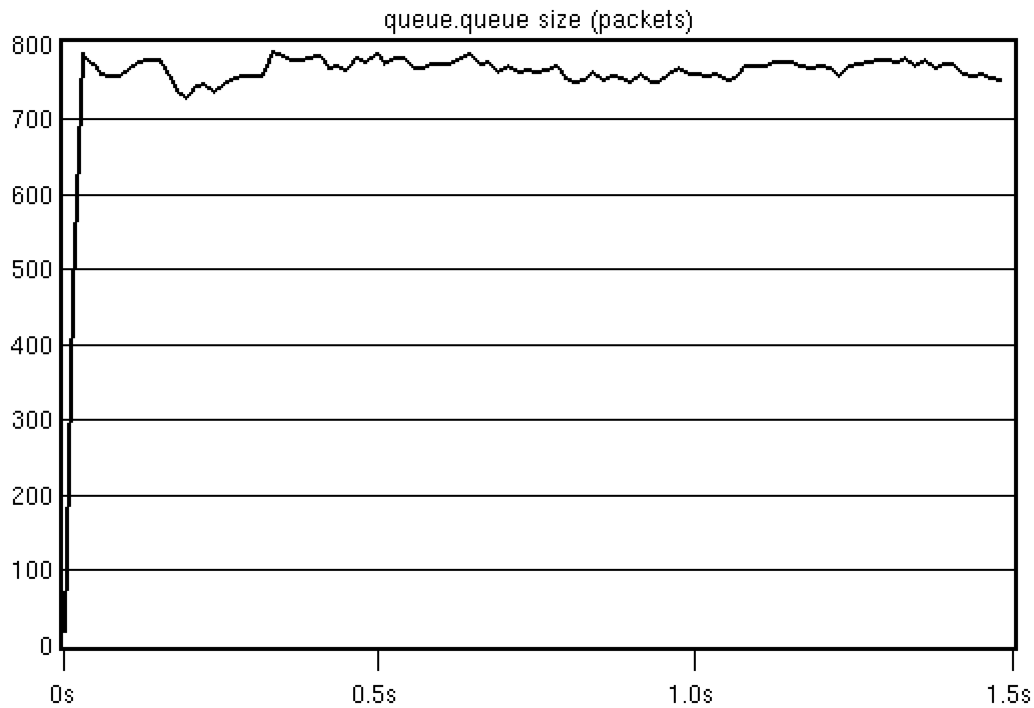

Figure 4.11 Switch 0 ABR queue length

From Figure 4.13 we can see that the throughput for the last switch not only is constant but also very close to 100% utilisation (97.4 %). This is very important since the controller not only avoids congestion but also fully utilises the available resources. Worth noting that it achieves this without changing the values of the design constants.

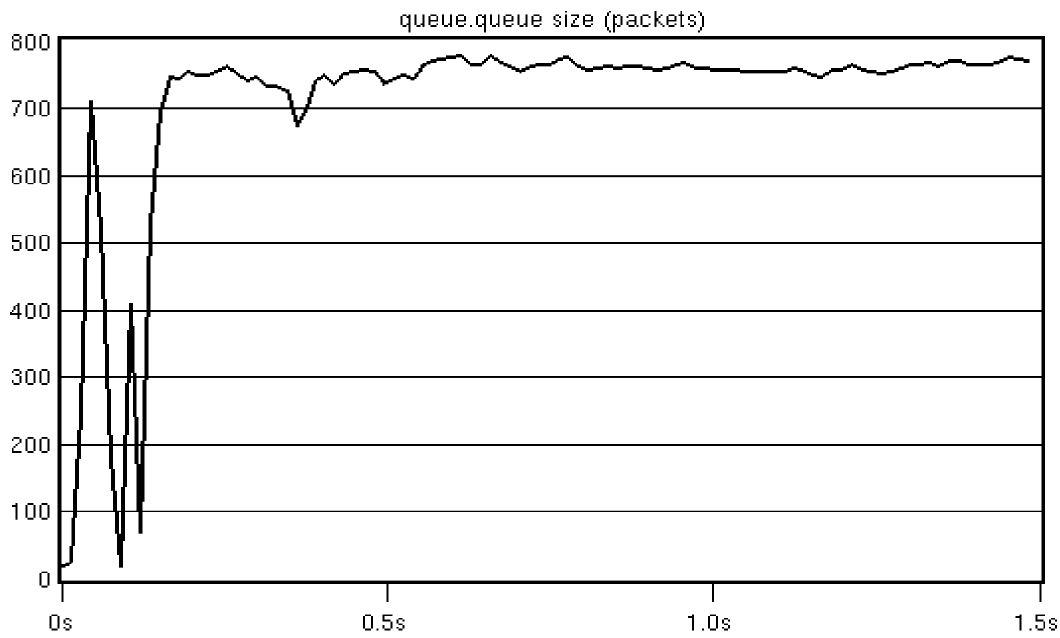

Figure 4.12 Switch 1 ABR queue length

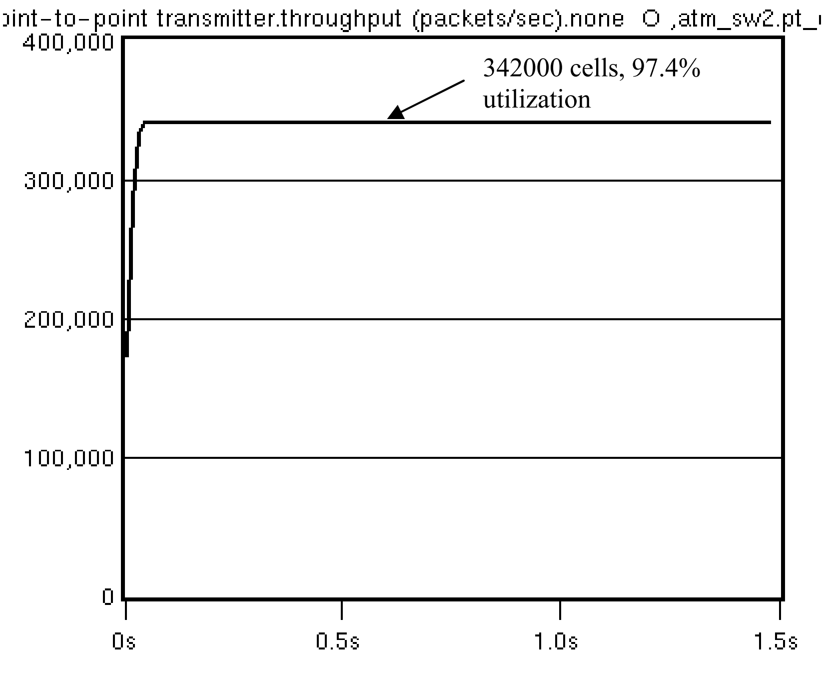

Figure 4.13 Switch 2 ABR queue length

Except utilisation and congestion avoidance, fairness is also very important. The controller not only must avoid congestion but also must share the available bandwidth in a fair manner. Fairness will not currently be investigated but remains an open issue for future work.

Chapter 5

Summary

5.1 Achievements

As we have seen most of the congestion control algorithms used today are based on ad hoc techniques and intuition and their effectiveness is proven through extensive simulations. The models proposed in chapter 4 are based on a simple fluid flow model, which uses non-linear control theory. We simulate these models in OPNET under different scenarios to show that they can offer satisfactory performance for control system designs. The results from the Dynamic Fluid Flow model show us that the behaviour of our model is very close to an event based real world network environment simulated using OPNET. This supports our assertion that the proposed model can adequately describe the dynamics of real networks. Based on this assertion, we defined a non-linear congestion controller using this model. The simulative performance evaluation results were very encouraging to pursue this further. Therefore we proposed a new strategy for congestion control: IDCC. The early performance results given by the IDCC congestion controller are very promising and encouraging for future work towards this direction. It is shown that non-linear congestion control can achieve controlled performance as dictated by the reference values. The advantage of such control design is that formal analysis of the behaviour can be carried out. This analysis can highlight any bounds in the achievable performance and provide guidelines for any heuristic that may be necessary.

5.2 Future Directions

Since feedback control is developed to be able to handle significant modelling errors and inaccuracies it is not surprising that preliminary results based on these models were very successful and encouraging for pursuing this approach further. It is safe to state that models which use non-linear control theory can be suitable for congestion control and that this seems to be the area we must concentrate our efforts.

Early results given by the IDCC model are very promising and encourage us to investigate this model further to see if it can be proposed as a more general congestion control scheme suitable for ISN networks. Further analysis and formal evaluation of the model is required. This means that the following steps must be done:

- formal analysis and evaluation of the model
- proof of fairness
- design of general and comprehensive scenarios
- implementation in multiple suite of networks (e.g. ATM, IP Diff-Serv)
- extension to large scale
- extensive simulations and evaluation of results

IDCC is a good starting point for further research.

A. Abbreviations

ABR	Available Bit Rate
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BECN	Backward Explicit Congestion Notification
B-ISDN	Broadband ISDN
CAC	Connection Admission Control
CAPC	Congestion Avoidance using Proportional Control
CBR	Constant Bit Rate
CLR	Cell Loss Ratio
Diff-Serv	Differentiated Services (protocol)
DS	Differentiated Services (field)
EF	Expedited Forwarding
EFCI	Explicit Forward Congestion Identification
EPRCA	Enhanced Proportional Rate Control Algorithm
ER	Explicit Rate
ERICA	Explicit Rate Indication for Congestion Avoidance
FERM	Fuzzy Explicit Rate Marking
FIFO	First In First Out
GCRA	Generic Cell Rate Algorithm
GFC	Generic Flow Control
HDTV	High Definition TV
IETF	Internet Engineering Task Force
IDCC	Integrated Dynamic Congestion Controller
IntServ	Integrated Services
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISN	Integrated Services Network

LAN	Local Area Network
N-ISDN	Narrow-band ISDN
PHB	Per Hop Behaviour
QoS	Quality of Service
RED	Random Early Discard
RIO	RED with In/Out
RM	Resource Management
RSVP	Resource ReSerVation Protocol
RTT	Round Trip Time
SLA	Service Level Agreement
SVC	Switched Virtual Connection
ToS	Type of Service
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UPC	Usage Parameter Control
VBR	Variable Bit Rate
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

B. Simulation model

In this appendix the Matlab and Opnet implementation models for the Dynamic Fluid Flow model and the non-linear congestion controller are presented. Some examples of OPNET source code for the IDCC model are also presented. Section B.1 presents in detail all the matlab models necessary to implement the non-linear congestion controller. In Section B.2 we present the process models for the ABR sources, the switch queue and the controller. Some sample C-like code from the Opnet implementation of the non-linear congestion controller and the IDCC controller, follows. Also there is some C-like code on how to implement the variable queue limit in the Guaranteed and Available traffic rate queues. The code presented gives an idea of the controller simplicity and implementation.

B.1 Matlab models for Dynamic Fluid Flow model and Non-linear congestion controller

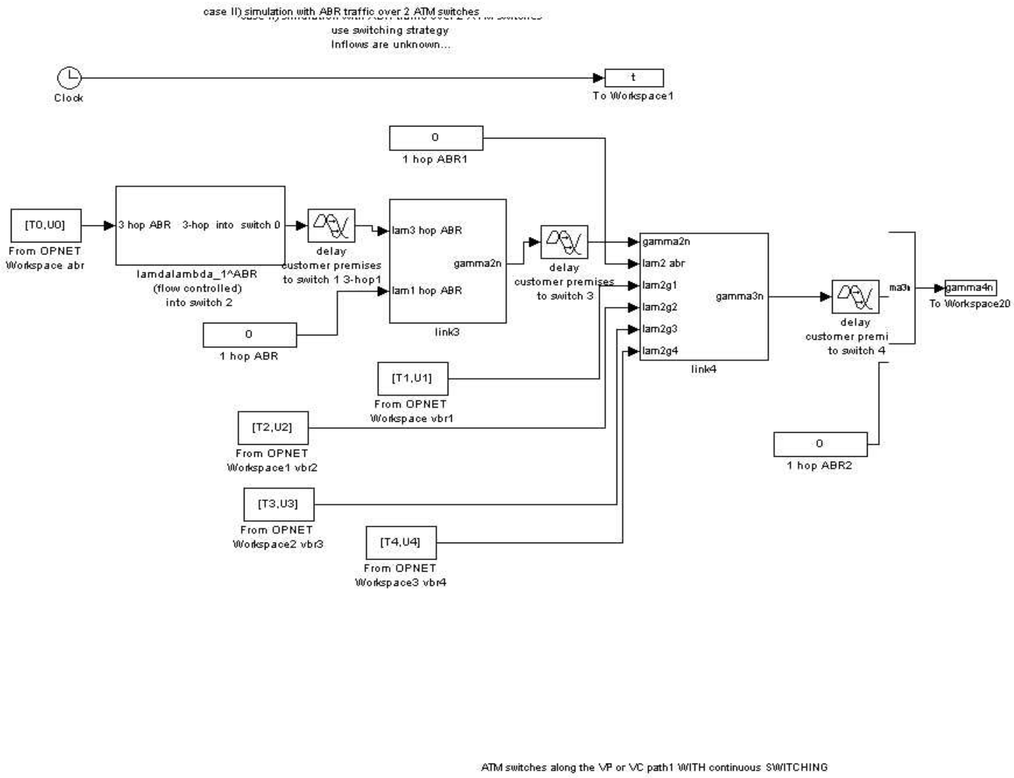

Figure B.1.1 Matlab main model

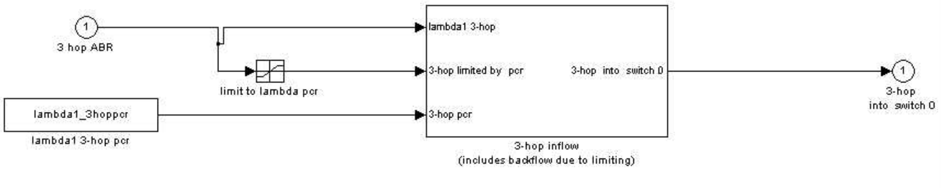

Figure B.1.2 Lamba ABR

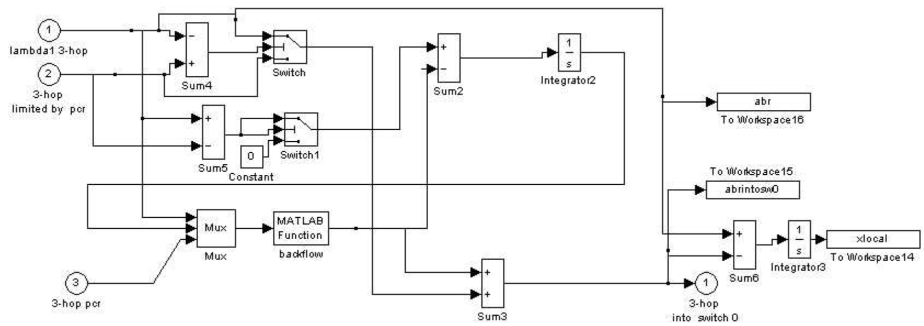

Figure B.1.3 3-hop inflow

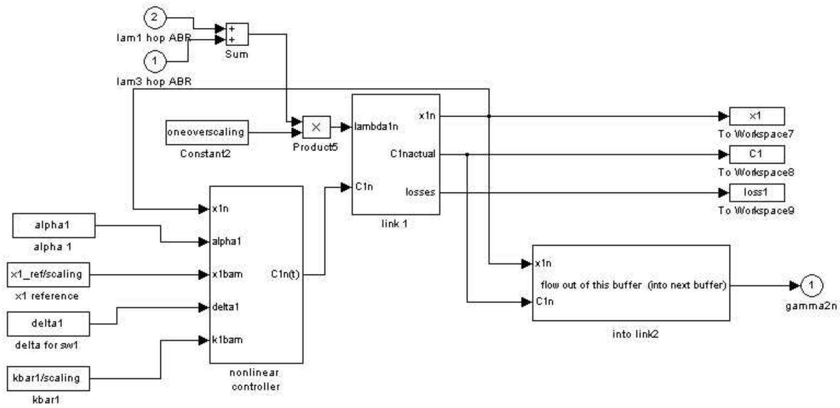

Figure B.1.4 Link 3 from main model

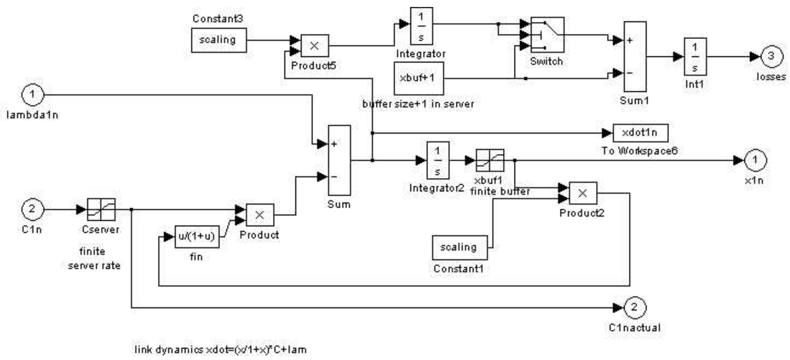

Figure B.1.5 Link1 of Link3 model

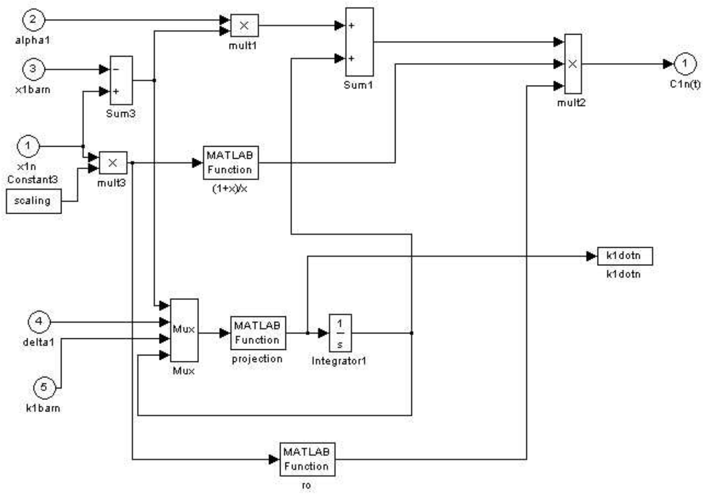

Figure B.1.6 Non-linear controller of Link3 model

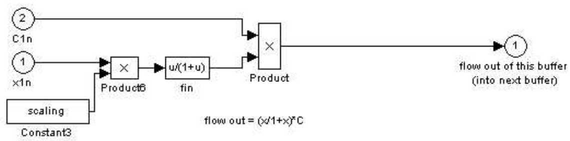

Figure B.1.7 Into Link 2 model of Link3 model

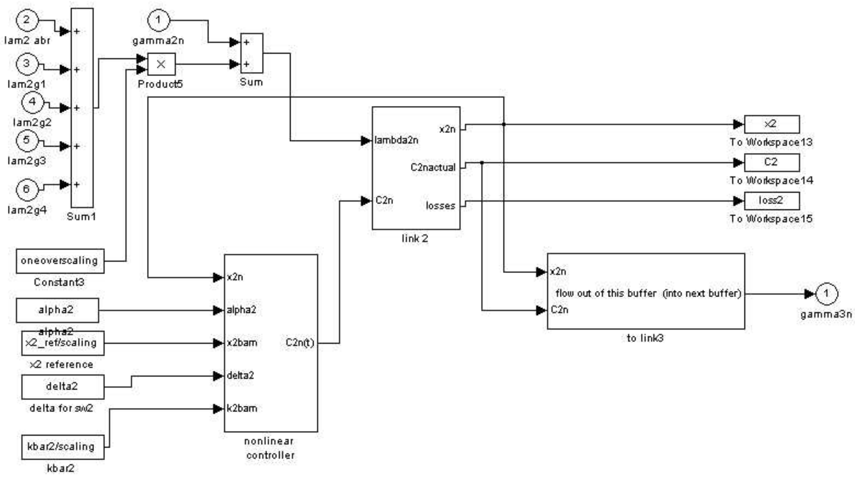

Figure B.1.8 Link 4 of main model

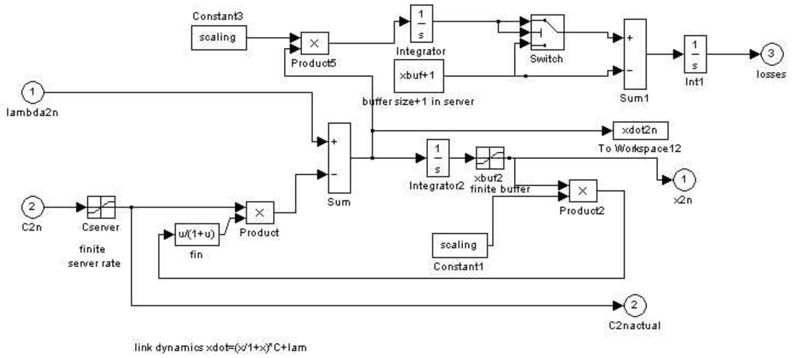

Figure B.1.9 Link2 of Link4 model

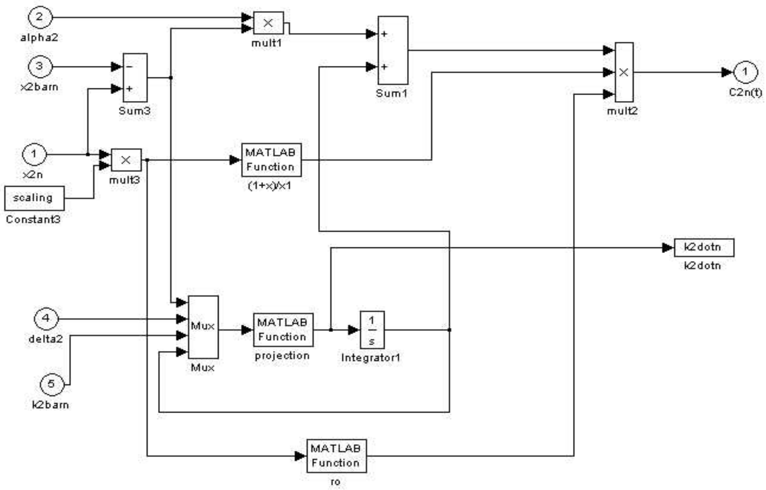

Figure B.1.10 Non-linear controller of Link4 model

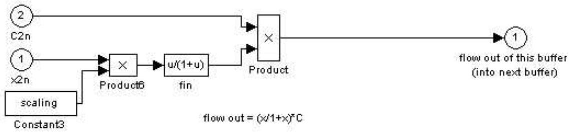

Figure B.1.11 To link3 of Link4 model

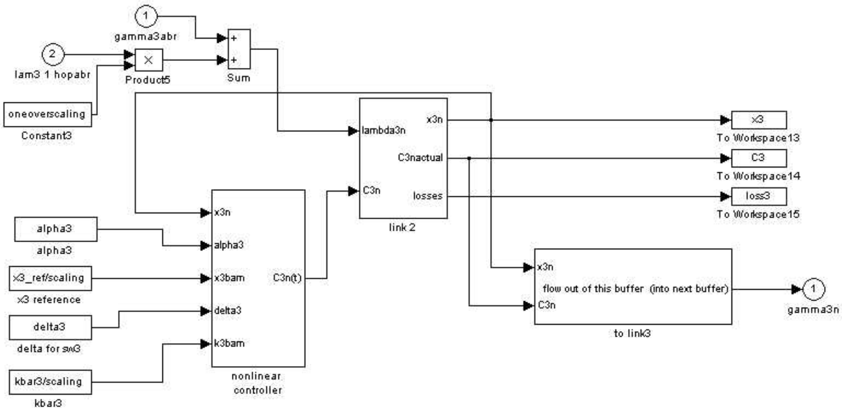

Figure B.1.12 Link 5 of main model

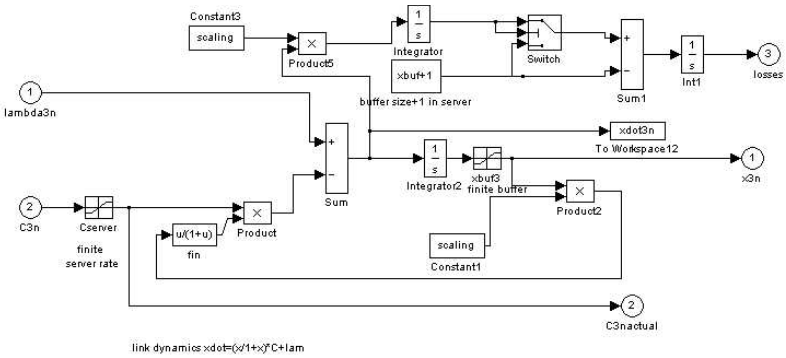

Figure B.1.13 Link 2 of Link 5 model

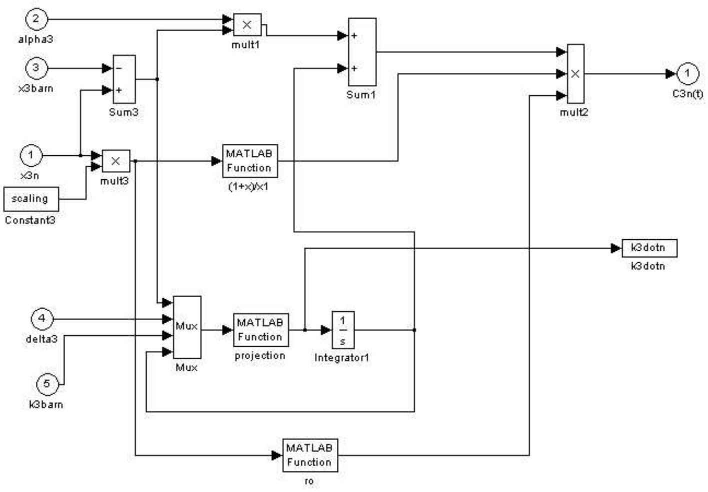

Figure B.1.14 Non-linear controller of Link 5 model

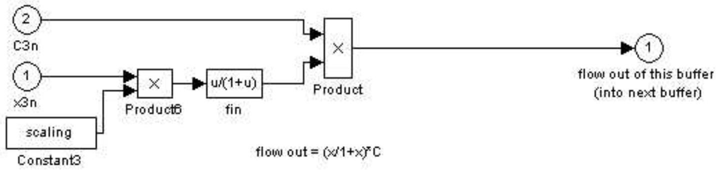

Figure B.1.15 To link3 of Link 5 model

B.2 Opnet models for Dynamic Fluid Flow model and Non-linear congestion controller

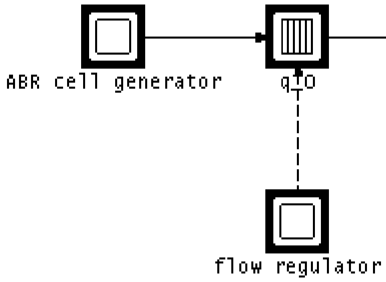

Figure B.2.1 ABR Source model

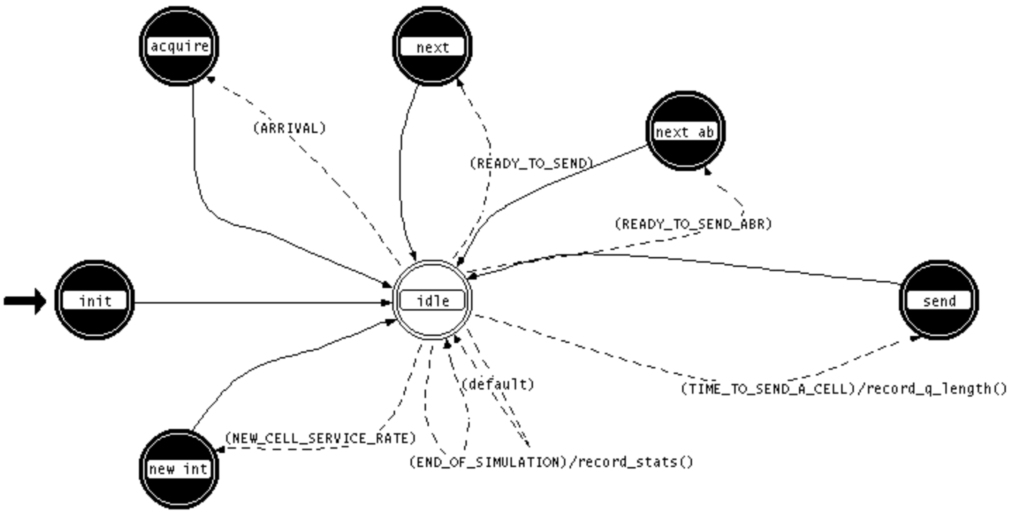

Figure B.2.2 Switch Queue model

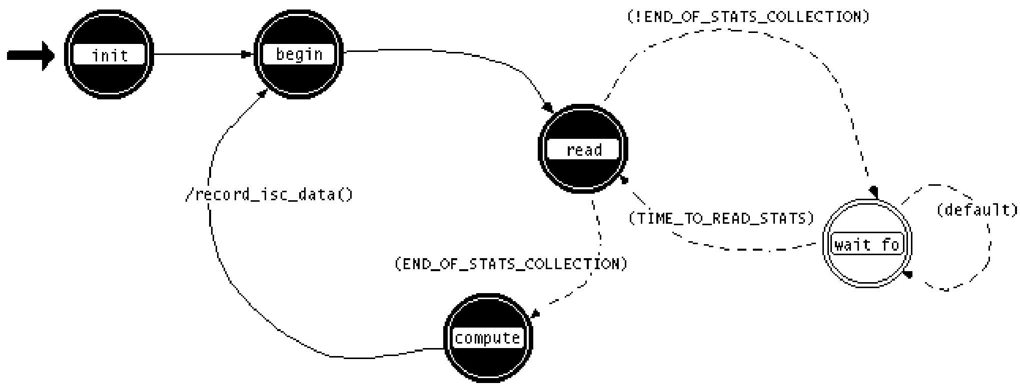

Figure B.2.3 Controller computation process

C - Code for congestion controller (compute process of Figure B.2.3)

```
/* Control; Strategy                              */
/* C=p *[( 1+x)/x]*[a*x^ + lamda(abr) + lamda(g)] */
/* where x is current queue length, a and x^ are constants */

/* Set p1 */
if (sv_avg_q_length <= 0.01) {
 sv_p1=0;
} /* if */
else { if ((sv_avg_q_length <= 1) & (sv_avg_q_length > 0.01)) {
          sv_p1 = ((100/99*sv_avg_q_length) - (100/99-1));
       } /* end if */
       else {  if (sv_avg_q_length > 1)
                sv_p1=1;
       } /* end else */
} /* end else */

/* Set x_kapello */
sv_x_kapello = sv_avg_q_length - sv_x_bar;


/* Calculate C1 */
if (sv_p1 == 0 ) {
  sv_cell_service_rate = 0;
}   /* end if */
 else {
sv_cell_service_rate = (sv_p1*((1 + sv_avg_q_length)/sv_avg_q_length))*
                        ((sv_alpha1*sv_x_kapello) + sv_abr_cell_arrival_rate +
                         sv_vbr_cell_arrival_rate );
}  /* end else */


sv_cell_service_rate = 0.8 / CELL_SERVICE_TIME;
```

```
if (sv_cell_service_rate < 0) {
  sv_cell_service_rate = 0;
}

/* hard limiter for cell service rate, it is a hack, */
/* we will change this later...                       */

if (sv_cell_service_rate > (1.0 / CELL_SERVICE_TIME)) {
  sv_cell_service_rate = 1.0 / CELL_SERVICE_TIME;
} /* if */
```

C - code for setting the reference value of the controller for Guaranteed and Available Traffic (begin process of figure B.2.3)

<u>Set reference point for Available traffic</u>

```
time=op_sim_time();
/* Set variable queue reference point */
if (time < 0.5) x_bar=900;
else { if (time < 1.0) {x_bar=300;}
       else {x_bar=600;}
      }
sv_q_length_totals = 0.0;
sv_stats_read = 0;
sv_gamma=100;
```

<u>Set reference point for Guaranteed traffic</u>

```
time=op_sim_time();
/* Set variable queue reference point */
if (time < 0.4) sv_x_bar=100;
else { if (time < 0.8) sv_x_bar=50;
       else { sv_x_bar=100;   }
     }

sv_alpha1=20000.0;
sv_beta=1;
```

IDCC code for calculating cell service rate for Available Bit rate sources based on available bandwidth

```
/* -*-C-*- */
/* Read available bandwidth at the switch */
sv_available_flow_rate = op_stat_local_read(ABR_RATE_INSTAT);
```

```
/* Calculate average queue length */
sv_avg_q_length = sv_q_length_totals/ISS_FILTER_PERIOD;


/* Control; Strategy                    */
/* lambda abr=p *[(Cmax-C(t))*x/(1+x)-(a*(x-x_bar))]

/* Computation */
computation = (sv_available_flow_rate * ( sv_avg_q_length /
(1+sv_avg_q_length))) - (alpha * (sv_avg_q_length-x_bar));

if (sv_kappa >= 353208)
 { sv_gamma=0; sv_kappa = 353208/10;
 }
if (sv_kappa <= -353208/10)
 { sv_gamma=0; sv_kappa = 353208/10;
 }
sv_kappa = sv_kappa - sv_gamma*(sv_avg_q_length-x_bar);

/* Set p */
if ( computation+sv_kappa < ERR )
  p=0;
else p=1;


/* Calculate lamda_abr (which is sv_cell_service_rate) */
if (sv_avg_q_length < (x_bar/10))
 sv_cell_service_rate = 353208;
else sv_cell_service_rate = p * (computation + sv_kappa);

/* Limit ABR rate to [ 0.4 Mbits - ABR available ] */
if (sv_cell_service_rate > sv_available_flow_rate)
  sv_cell_service_rate = sv_available_flow_rate;
if (sv_cell_service_rate <= 1000.0)
  sv_cell_service_rate = 0.0;
```

C. References

[1] S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Transactions on Networking, pages 458-472, Vol. 7, No. 4, August 1999
[2] R. Braden et al, " Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification, RFC 2205, September 1997.
[3] L. Zhang et al., "RSVP: A new Resource ReSerVation Protocol", IEEE Network, pages 8-18, September 1993.
[4] S. Shenker et al, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.
[5] J Wroclawski, "Specification of Controlled-Load Network Element Service", RFC 2211, September 1997.
[6] D. Clark et al. "Integrated Services in the Internet Architecture: An Overview", July 1994, RFC 1633
[7] S. Blake et al. "An architecture for Differentiated Services", December 1998, RFC 2475
[8] K. Nichols et al. "Definition of the differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers, December 1998, RFC 2474
[9] V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB", February 1999, draft-ietf-diffserv-phb-ef-02.txt
[10] J. Heinanen et al. "Assured Forwarding PHB Group", February 1999, draft-ietf-diffserv-af-06.txt
[11] ATM Forum, ATM Forum Traffic Management Specification, Version 4.0, af-tm-0056.000, April 1996
[12] P. Newman, "Backward explicit congestion notification for ATM local area networks", GLOBECOM'93, Houston, USA, December, 1993
[13] L. Roberts, "Enhanced PRCA (Proportional rate control algorithm)", Tech. Rep. AF-TM 94-0735R1, August 1994.
[14] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, "ERICA switch algorithm: a complete description", ATM Forum, AF/96-1172, August 1996
[15] C.E. Rohrs and R. A. Berry and S. J.O'Halek, A Control Engineer's Look at ATM Congestion Avoidance, IEEE Global Telecommunications Conference GLOBECOM'95, Singapore, 1995.
[16] N. Mitrou, Traffic control in ATM: A review, an engineer's critical view and a novel approach, Journal of COMPUTER NETWORKS & ISDN SYSTEMS, 1999
[17] D. McDysan, D. Spohn, ATM Theory and Applications, McGraw-Hill, 1998
[18] R. Jain et al, "Source Behaviour for ATM ABR Traffic Management: An Explanation", IEEE Communications 1996.
[19] F. Bonomi, K. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service", IEEE Network, March/April 1995.
[20] H. T. Kung, R. Morris, "Credit-based Flow Control for ATM Networks", IEEE Network, March/April 1995.
[21] V. Jacobson, Congestion Avoidance and Control, ACM SIGCOMM88, 1988.
[22] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, January 1997.
[23] P. Karn, C. Partridge, ``Improving Round-Trip Time Estimates in Reliable Transport Protocol", ACM SIGCOMM87, October, 1987.

[24] W. Stevens, TCP/IP Illustrated, Volume 1 The Protocols', Addison-Wesley, 1994.
[25] V. Jacobson, R. Braden, D. Borman, TCP Extensions for High Performance, RFC 1323, May 1992.
[26] K.K. Ramakrishnan, and S. Floyd, A proposal to add explicit congestion notification (ECN) to IP, draft-kksjf-ecn-03.txt, October 1998. (RFC2481, January 1999).
[27] Braden et al, Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC2309, April 1998.
[28] K.K. Ramakrishnan, Bruce Davie, Sally Floyd, "A Proposal to Incorporate ECN in MPLS", internet-draft, http://www.aciri.org/floyd/papers/draft-mpls-ecn-00.txt, July 1999.
[29] L. Roberts, "Enhanced PRCA (Proportional rate control algorithm)", Tech. Rep. AF-TM 94-0735R1, August 1994.
[30] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, ERICA switch algorithm; a complete description, ATM FORUM, AF/96-1172, August 1996.
[31] C.E. Rohrs and R. A. Berry and S. J.O'Halek, A Control Engineer's Look at ATM Congestion Avoidance, IEEE Global Telecommunications Conference GLOBECOM'95, Singapore, 1995.
[32] J. Martin, A. Nilsson, The evolution of congestion control in TCP/IP: from reactive windows to preventive flow control, CACC Technical Report TR-97/11, North Carolina State University, August 1997. (available on line http://www2.ncsu.edu/eos/ info/ece_info/www/ ccsp/tech_reports/ faculty/ nilsson.html).
[33] T.V. Lakshman and U. Madhow, The Performance of TCP/IP for Networks with High Bandwidth Delay Products and Random Loss, IEEE/ACM Transactions on Networking, vol. 5, pp. 336-350, June 1997.
[34] A.Feldmann, A.C. Gilbert, W. Willinger, "Data Networks as cascades: Investigating the multifractal nature of the Internet WAN traffic", SIGCOMM 98, Vancouver, 1998.
[35] A. Feldmann, P. Huang, A.C. Gilbert, W. Willinger, Dynamics of IP traffic: A study of the role of variability and the impact of control, To appear at ACM/SIGCOMM 1999.
[36] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", ACM/IEEE Transaction on Networking, 1(4): pgs 397-413, August 1993
[37] Wu-chang Feng, "Improving Internet Congestion Control and Queue Management Algorithms", PhD Dissertation, 1999
[38] A. Segall, "The modelling of adaptive routing in data communication networks", IEEE Transactions on Communications, Vol. 25, No.1, pp. 85-95, January 1977.
[39] S. Keshav, "A control theoretic approach to flow control", ACM SIGCOMM'91, Zurich, Switzerland, 1991.
[40] L. Benmohamed, Y.T. Yang, "A Control-Theoretic ABR Explicit Rate Algorithm for ATM Switches with Per-VC Queuing", Infocom 98, 1998.
[41] A. Pitsillides, J. Lambert, "Adaptive connection admission and flow control: quality of service with high utilization", IEEE INFOCOM'94, 13th Conference on Computer Communications, Toronto, Ontario, Canada, June 1994, pp.1083-1091.

[42] A. Pitsillides, J. Lambert, "Adaptive congestion control in ATM based networks: quality of service with high utilization", Journal of Computer Communications, 20, 1997, pp. 1239-1258.
[43] A. Pitsillides, A. Sekercioglu, G. Ramamurthy, "Fuzzy backward congestion notification (FBCN) congestion control in asynchronous transfer mode (ATM) networks", IEEE GLOBECOM'95, Global Telecommunications Conference, Singapore, November 13-17, 1995, pp. 280-285.
[44] A. Pitsillides, A. Sekercioglu, G. Ramamurthy, "Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking (FERM)", IEEE JSAC, Volume 15, Issue 2, February 1997, pp. 209-225.
[45] Y.C. Liu and C. Douligeris, Rate Regulation with Feedback Controller in ATM Networks-A Neural Network Approach, IEEE JSAC, vol. 15, no.2, Feb. 1997, pp.200-208.
[46] A. Pitsillides, A. Sekercioglu, Fuzzy Logic based effective congestion control, TCD workshop on Applications of Computational Intelligence to Telecommunications, London, 16 May, 1999.
[47] A. Pitsillides, A. Sekercioglu, A. Vasilakos, Computational Intelligence in management of ATM networks: A survey of current state of research, ESIT'99, European Symp. of Intelligent Techniques, June 3-4, Crete, Greece, 3-4 June 1999.
[48] A. Pitsillides, P. Ioannou, "Combined nonlinear control of flow rate and bandwidth for virtual paths in ATM based networks", 3rd IEEE Mediterranean Symposium on new directions in control and automation", Limassol, Cyprus, July 11-13 1995, pp.177-184.
[49] A. Pitsillides, P. Ioannou, D. Tipper, "Integrated control of connection admission, flow rate, and bandwidth for ATM based networks", IEEE INFOCOM'96, 15th Conference on Computer Communications, San Francisco, USA, March 24-28, 1996, pp. 785-793.
[50] A. Pitsillides, P. Ioannou, D. Tipper, "Integrated control of connection admission, flow rate, and bandwidth for ATM based networks: A comparative evaluation", ICT'96, International Conference on Telecommunications, Istanbul, Turkey, 14-17 April 1996, pp.504-507.
[51] A. Pitsillides, P. Ioannou, "An Integrated Switching Strategy for ABR Traffic Control in ATM Networks" ISCC'97, IEEE Symposium on Computers and Communications, Alexandria, Egypt, 1-3 July 1997.
[52] A. Sekercioglu, A. Pitsillides, P. Ioannou, "A Simulation Study on the Performance of Integrated Switching Strategy for Traffic Management in ATM Networks" ISCC'98, IEEE Symposium on Computers and Communications, Athens, Greece, 30 June – 2 July 1998, pp. 13-18.
[53] A. Isidori, "Nonlinear Control Systems", 2nd Ed. New York, Springler-Verlag, 1989.
[54] C.I. Byrnes, A. Isidori, and J.C. Willems, "Passivity, feedback equivalence, and the global stabilisation of minimum phase nonlinear systems, IEEE Tr. Automatic Control, vol. 36, 1991, pp. 1228-1240.
[55] Z. Arstein, "Stabilization with relaxed controls", Nonlinear Anal., 1983, pp. 1163-1173.
[56] R.A. Freeman, P.V. Kokotovic, "Inverse optimality in robust stabilisation", SIAM J. Control and Optimisation, vol. 34, no. 4, July 1996, pp. 1365-1391.

[57] M. Krstic, I. Kanallakopoulos, P.V. Kokotovic, “Nonlinear and Adaptive Control Design”, John Wiley&Sons, 1995.
[58] K.S. Narendra, S. Mukhopadhaay, “Adaptive control using neural networks and approximate models”, IEEE Transactions On Neural Networks, vol. 8, 1997, pp. 475-485.
[59] M.M. Polycarpou and P.A. Ioannou, “Identification and control of nonlinear systems using neural network models: design and stability analysis”, Tech. Rep. 91-09-01, University of Southern California, September 1991.
[60] R. Ordonez, J. Zumberge, J.T. Spooner, “Adaptive fuzzy control: experiments and comparative analysis, IEEE Transactions of Fuzzy Systems, vol. 5, 1997, pp. 167-188.
[61] P.A. Ioannou and J. Sun, Stable and robust Adaptive Control, Englewood Cliffs, NJ: Prentice Hall, 1996.
[62] A. Datta and P.A. Ioannou, Performance improvement versus robust stability in model reference adaptive control, IEEE Transactions on Automatic Control, vol. 49, pp. 2370-2388, 1994.
[63] A. Datta and P.A. Ioannou, Directly computable L2 and Loo bounds for Morses dynamic certainty equivalence adaptive controller, International Journal of Adaptive Control and Signal Processing, vol. 9, pp. 423-432, 1995.
[64] D. Tipper, M. K. Sundareshan, "Numerical Methods for modelling Computer Networks Under Nonstationary Conditions", IEEE JSAC, Dec. 1990.
[65] C. Agnew, Dynamic modelling and control of congestion-prone systems, Operations Research, Vol. 24, No. 3, 1976, pp. 400-419.
[66] J. Filipiak, "Modelling and Control of Dynamic Flows in Communication Networks", Springer-Verlag, 1988.
[67] S. Sharma, D. Tipper, "Approximate models for the study of nonstationary queues and their application to communication networks", IEEE ICC'93, May 1993.
[68] X. Gu, K. Sohraby, D. R. Vaman, “Control and performance in packet, circuit, and ATM networks”, Kluwer Academic Publishers, 1995.
[69] D. Anick, D. Mitra and M.M. Sondhi, "Stochastic Theory of a Data-Handling System with Multiple Sources", The Bell System Technical Journal, Vol. 61, pp. 1871-1894, 1982.
[70] H. Kobayashi and Q. Ren, "A Mathematical Theory for Transient Analysis of Communications Networks", IEICE Transactions on Communications, Vol. E75-B, pp. 1266-1276, 1992.
[71] Y. Wardi, B. Melamed, "Comtinuous Flow Models: Modeling, Simulation and Continuity Properties", Proceedings of the 38th Conference on Decision & Control, Phoenix, Arizona, USA, December 1999.
[72] T. Oser, X. Gu, D. R. Vaman, “Routing with admission control in ATM networks”, Journal of Network Systems and Management, vol. 3, no. 3, 1995.
[73] M. Gerla, J. A. S. Monteiro, R. Pazos, "Topology Design and Bandwidth Allocation in ATM nets", IEEE JSAC, Vol. 7, No. 8, Oct. 1989, pp. 1253-1262.
[74]L. Rossides, A. Pitsillides, P. Ioannou, "Non-linear Congestion control: Comparison of a fluid flow based model with OPNET simulated ATM switch model", TR-99-1, University of Cyprus, 1999.
[75] OPNET Modeller (OPtimized Network Engineering Tools), Mil 3, Inc, Washington DC.

[76] J-C Bolot, A.U. Shankar, Analysis of a fluid approximation to flow control dynamics, IEEE Infocom 92, Florence, Italy, May 1992, pp. 2398-2407.
[77] B. Vandalore, R. Jain, R. Goyal, S. Hahmy, Design and analysis of queue control functions for explicit rate switch schemes, IC3N'8, Lafayette, Louisiana, October 18, pp. 780-786.
[78] K. Kawahara, Y. Oie, M. Murata, H. Miyahara, Performance analysis of reactive congestion control for ATM networks, IEEE Journal of Selected Areas in Communications, Vol. 13, No. 4, May 1995, pp. 651-661.
[79] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, J. Robbins, +ACI-Performance models of statistical multiplexing in packet video communications+ACI-, IEEE Transactions on Communications, Vol. 36, No. 7, pp. 834-844, July 1988.