Congestion Control in Networked Systems

Yong Jun YU

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science

Concordia University

Montreal, Quebec, Canada

September 2005

Your file Votre référence
ISBN: 978-0-494-16248-4
Our file Notre référence
ISBN: 978-0-494-16248-4

Acknowledgements

I would like to express my sincere appreciation to Dr. Kash Khorasani of Concordia University, who introduced me to the area of congestion control in networked systems including the subjects of this thesis. All the work conducted in the context of this thesis would not have been possible without his support and advice. Thanks are also due to my friend, Tao Jiang, Yang Hong, who gave me insightful assistance and helpful comments and feedback. Finally, I am deeply grateful to my parents, for their understanding and support during the elaboration of this thesis.

Abstract

This thesis discusses the centralized and decentralized congestion control strategies in differentiated services of a networked system. First a sensor-decision maker-actuator (3 nodes) network structure is designed and traffic flow is divided into three classes: Premium, Ordinary and Best effort. For each node we design a control strategy based on the dynamic fluid flow model and M/M/1 queuing theory to allocate bandwidth to different class traffic for avoiding congestion. For the whole network, we design control algorithms based on both centralized and decentralized methods. The stability of the closed loop system is analyzed and some comparisons between centralized and decentralized schemes are made. Several examples and simulation results are included to illustrate the design methodology.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction to networked control systems

Networked control systems are systems whose sensors, actuators, and decision makers (controller units) are connected through communication networks [1]-[6]. Information or data among sensors, decision makers and actuators must be exchanged over a communication network. Networked systems have become an enabling technology for many military, commercial and industrial applications.

This type of system has the advantage of greater flexibility with respect to traditional control systems. Also, it allows for reduced wiring, as well as a lower installation cost. It also permits greater ability in diagnosis and maintenance procedures. Examples of such systems can be seen in UAV (Unmanned Aerial Vehicle) [7], UGV (Unmanned Ground Vehicle), UUV (Unmanned Underwater Vehicle) or manufacturing plants.

An example of a networked system is shown in Figure 1-1 for a UAV's fault tolerant system, where we want to trace the location of the target (error) and take corresponding

actions. So we use a group of sensors to detect the target's location, then send information to decision makers. Based on this information, decision makers will send signals to the actuators to take actions.

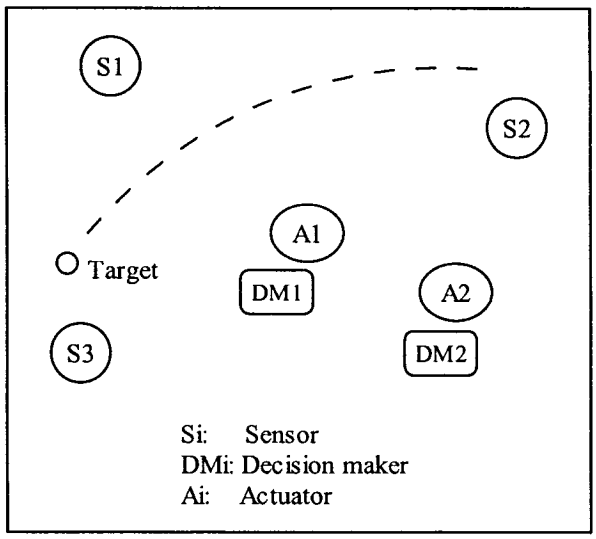

Figure 1-1: An example of networked control system

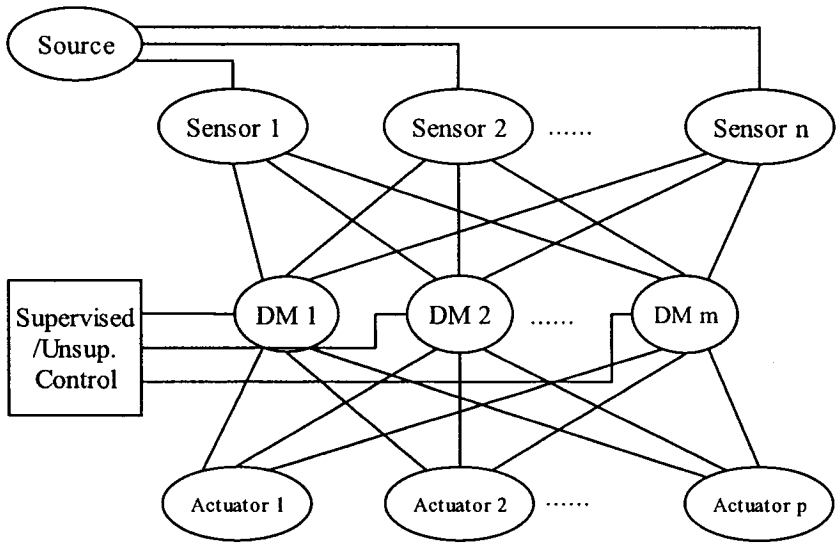

Figure 1-2: Structure of a networked control system

Figure 1-2 gives us a more detailed structure of such a network. It could be a supervised system, which controller can give instructions to each decision maker to do specific operation. Also, in an unsupervised system, these decision makers will control the whole system without any instruction.

Networked devices include sensors, actuators and networked controllers. Sensors have three major features: data acquisition, intelligence and communication ability [8][9]. Sensors acquire proper physical data such as speed and temperature from the practical environments and have an application processor which is the interface between the sensor and network. Similar to sensors, actuators have the functions of actuation, intelligence and communication [10]. The actuator should be able to decode information from the decision makers and transmit it into the physical devices. Besides network-capable application processors, the major functionalities of networked controllers are to analyze the sensor data, make decisions, and give commands to actuation devices.

For more specific and basic unit, we study the single Sensor-Decision Maker-Actuator (S-DM-A) channel. The structure of a single S-DM-A channel is shown in Figure 1-3. Usually, we need feedback information to monitor the action of each unit. Sensor receives information or data from outside, decision maker and actuator, then sends data to decision maker. Decision maker makes decisions by analyzing the data sent by sensor and actuator, and then gives commands to actuator and feedback information to sensor. Actuator will act following decision maker's instruction, and give the result (information) to decision maker and sensor.

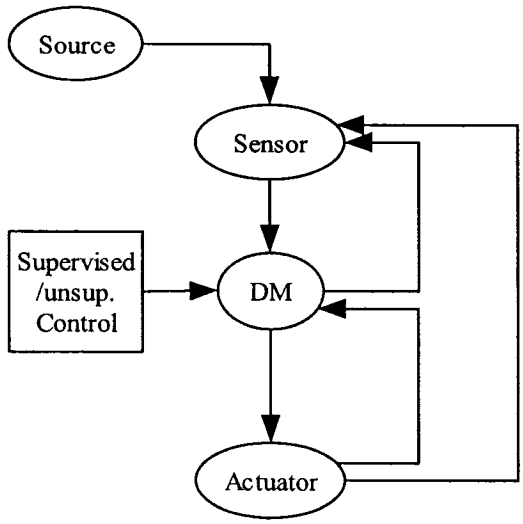

Figure 1-3: Structure of single S-DM-A channel

1.2 Congestion control in networked systems

1.2.1 Introduction to congestion control

Congestion is defined as a state of network elements (e.g. switches, concentrators, cross-connects and transmission links) in which the network is not able to meet the negotiated network performance objectives for the already established connections and/or for the new connection requests [11]. Congestion control refers to the set of actions taken by the network to minimize the intensity, spread and duration of congestion.

Congestion is caused by saturation of network resources (e.g., communication links, buffers, switches...). In a real network, many aspects that could cause congestion should be considered, for example, transmit delay, transmit protocol and link capacity. The

resources (transmitting medium, protocols) are limited, however, users and their demands are exponentially increasing. If data is sent at excessive rates, that is, arrival rate is greater than the service rate, then congestion occurs. For example, if a network link delivers packets to a queue at a higher rate than the service rate of the queue, then the size of the queue will grow. In real circumstances, the queue size and waiting buffer are of finite size, and therefore packets will be lost. The network throughput will dramatically decrease and much information will be dropped (as shown in Figure 1-4).

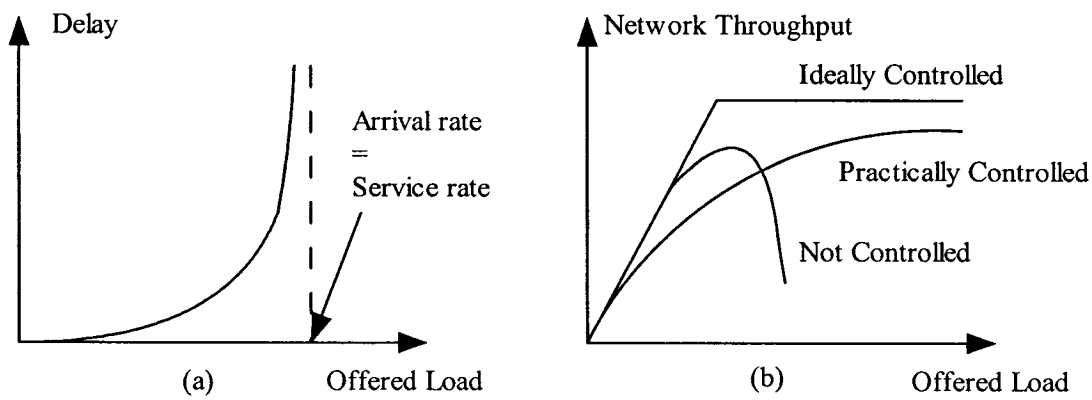

Figure 1-4: Network delay and throughput versus offered load

Because of the limited resources, we want to maximize our network throughput. So we need control and allocate the network resources. Congestion is not a static resource shortage problem, but rather a dynamic resource allocation problem.

1.2.2 Congestion control in networked systems

Unlike regular control systems, in networked control systems the synchronization

between different sensors, actuators and decision makers is not guaranteed. Furthermore, there is no guarantee for zero delay or even constant delay in sending information from sensors to the decision makers and control signals from the decision makers to the actuators. When there is congestion in the communication network, some packets are dropped to either reduce the queue size in the path or to inform the senders to reduce their transmission rates. In real-time systems, particularly control systems, delays or dropped packets may be catastrophic and may cause instability in the control systems.

Despite the many years of research efforts, the problem of network congestion control remains a critical issue and a high priority, especially given the growing size, demand, and capacity (bandwidth) of the network. Network congestion is becoming a real threat to the growth of existing data networks, and to the future deployment of integrated services communication networks. Networks will service all users' requests, however, their resources are limited, so congestion will occur. It is an important problem that cannot be ignored. Thus there must be a mechanism to regulate those resources to different users to avoid congestion.

Figure 1-5 shows the traffic flow of the single S-DM-A channel. Between every two nodes, we define different gain of the link and different delay of transmission in each link (details will be discussed in Chapter 4).

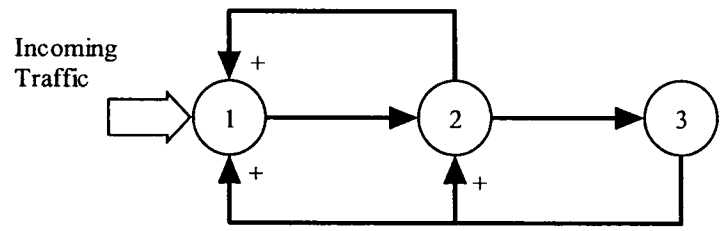

Figure 1-5: Traffic flow in the single S-DM-A channel

According to Figures 1-2 and 1-3, we know that all nodes (sensor or decision maker or actuator) are almost fully connected, so in Figure 1-5, the incoming traffic to a node is a summary of incoming flow, which means the incoming traffic is the summary of traffic from sensors. However, the link capacity cannot be infinite, so the information cannot be fully served when the incoming flow is heavy. Under this situation, congestion will occur, which results in packets dropping and message being lost. So we must take measures to avoid congestion [12], [13], [14], for example, use the Diff-Serv architecture [15], [16] to distinguish the high priority messages from the traffic, and serve them first. In practical environment, different incoming traffic holds different priority that is defined by the user, so it's necessary to serve the highest ones first. That is, to assure the highest priority traffic gets enough bandwidth to pass through the channel. Therefore, the problem becomes how to allocate bandwidth to each traffic channel according to different priorities to avoid congestion.

The need for new technology and network independent, congestion control algorithms, are more demanding than ever. To satisfy these needs we must consider a new totally different approach based not necessarily on classical queuing theory. There are mainly two types of concepts of avoiding congestion. One is to allocate bandwidth to different

users by a fair mechanism [17] and another one is by priorities [15], [16]. We mainly consider and are concerned about bandwidth allocation for one user. However, for different services with different priorities, which is similar to the bandwidth allocation among multi users with different priorities, user who holds the highest priority should be served first, then the lower ones, and finally the user of least priority.

1.3 Current trends and contributions of the thesis

Most of the congestion control algorithms used currently are based on ad-hoc techniques and intuition and their effectiveness is proven through extensive simulations. Recently fluid flow model is widely used in the design of congestion algorithms. Based on this, many researchers have done a lot work in this field, for example, flow rate control [17], [18], [19], queue length control [16], [20], and some special methods over TCP/IP [21] or ATM [22], [23].

Although fluid flow based model for congestion control has been extensively studied, the combination of it with non-linear control theory has not been extensively studied. This is probably because of the complexity of combination of the control theories and the existing models. Recently several attempts have been made to develop congestion controllers using optimal control theory [24]; linear control [14], [25], [26]; predictive adaptive control [27]; fuzzy and neural control [28], [29], [30]; and non-linear control [20], [31], [32], [33]. 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. The most important recent developments in non-linear control theory include feedback linearization [34], passivity theory [35], control Lyapunov functions [36], [37], back stepping and tuning functions [38], neural and fuzzy control systems [39], [40], [41], and robust adaptive control for linear and non-linear systems [42], [43], [44].

Our objective is to efficiently use the finite bus capacity while maintaining well closed loop control system performance, including stability, rise time, overshoot and other design criteria. 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. 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. Early attempts to use non-linear control theory for network congestion control include [20], [31], [32], [33]. The models are based on a fluid flow model, and use non-linear control theory [16], [45], [46], [47].

However, this method is only effective for a single node or a series of nodes without any feedback information. And it is also limited when the traffic flow is low. In this thesis, first, we have improved the above algorithms by applying a switching control to the condition when the traffic is low. When incoming traffic changes, the control algorithm will automatically switch from one to the other one according to the traffic.

Next we design a traffic flow structure of a three-node network (the structure is shown in Figure 1-5) which simulates the Sensor-Decision Maker-Actuator structure. There is feedback information from decision maker to sensor and from actuator to decision maker and to sensor. Based on this structure, we design both decentralized and centralized methods to regulate traffic flow with three different priorities (premium, ordinary and best effort) among three nodes (sensor – decision maker - actuator), and analyze the stability of the different algorithms. we simulate these models by Simulink and Matlab 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 behaviors of our model is very close to an event based real world network environment.

In chapter 2, general ideas of dynamic fluid flow model and M/M/1 queue are introduced, including the model, the algorithm and some simulation results. In chapter 3, we investigate and propose such congestion control schemes used for multi services in one node. In chapter 4, decentralized and centralized methods are implemented to regulate three types of traffic flow among three nodes; additionally, the stability of system is analyzed and comparisons are made. A conclusion is drawn in Chapter 5, and some future directions are also discussed.

Chapter 2

Dynamic Fluid Flow Model

2.1 Introduction

Fluid flow model is widely used to design model or simulate performance of network systems. Although fluid flow model used for congestion control has been extensively studied, the combination of it with non-linear control theory has not been widely studied. In this chapter, we introduce a congestion control scheme based on the fluid flow model and non-linear control theory [16], [45], [46], [47].

We derive the control strategy for a non-linear dynamic model of a network queue (node) based on the fluid flow considerations and matching the M/M/1 queue behavior at equilibrium. Although this model cannot accurately predict the behavior of a network system, more and more research and study to design congestion controllers using this model has been initiated because it captures the dominant dynamics of the network system, but neglects secondary effects and the noisy environment.

In the following two sections we will present a simple, dynamic fluid flow model that is

able to capture the essential dynamics of the network node. Based on this model we will design a simple non-linear congestion controller. The control strategy is to attempt to keep a queue buffer length close to a reference value without knowledge or measurement of the incoming traffic flow.

2.2 Fluid Flow model

2.2.1 Mathematical model

In this section a dynamic model is introduced, in a form suitable for a distributed control solution. The objective is to find a mathematical model, which captures the essential dynamic behavior, but has less complexity. Figure 2-1 shows a simple model of a network node.

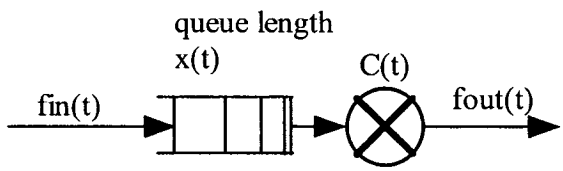

Figure 2-1: Model of a network node

For a single queue, using the flow conservation principle and assuming no losses, let $x(t)$ be a state variable denoting the ensemble average of the number of bits in the system at time t, furthermore, let $f_{out}(t)$ and $f_{in}(t)$ be the ensemble average of bits flow entering and leaving the system, respectively; 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:

$$\dot{x} = -f_{out}(t) + f_{in}(t) \tag{2-1}$$

The fluid flow equation is quite general and can model a wide range of queuing and contention systems. 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 utilization of the link $\rho(t)$ by $f_{out}(t) = C(t)\rho(t)$, where $C(t)$ is defined as the capacity of the queue server. We assume that $\rho(t)$ can be approximated by a function $G(x(t))$ which represents the ensemble average utilization 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 nonlinear differential equation of the form [15]:

$$\dot{x} = -G(x(t)) * C(t) + \lambda(t), \qquad x(0) = x_0 \tag{2-2}$$

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 (2-2) 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 [48], [49], [50]. Other approaches, such as system identification techniques and neural networks, can also be used to identify the parameters of the fluid flow equation.

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 [48]. 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

$$\dot{x} = -\frac{x(t)}{1+x(t)} * C(t) + \lambda(t), \qquad x(0) = x_0 \tag{2-3}$$

Note that this equation is valid for $0 \le x(t) \le x_{buffer\_size}$ and $0 \le C(t) \le C_{server}$ where X_{buffer_size} is the maximum possible queue size and C_{server} is the maximum possible server rate (usually the maximum link capacity).

2.2.2 Simulation results based on the fluid flow model

According to (2-3), we construct a fluid flow model by Simulink and Matlab to perform simulation. We chose the link capacity of the node as $C_{server} = 155$Mbits/sec and $x_{buffer\_size} = 5M bits$. We generate the incoming traffic as shown in Figure 2-2, which represents the rate of incoming traffic with the mean value of 10Mbits/sec and variance of 2Mbits/sec. Because there in no rules to regulate or evaluate the bandwidth allocated to the node, so we generate random value which approximates the incoming traffic.

Figure 2-3 shows the bandwidth allocated to the node.

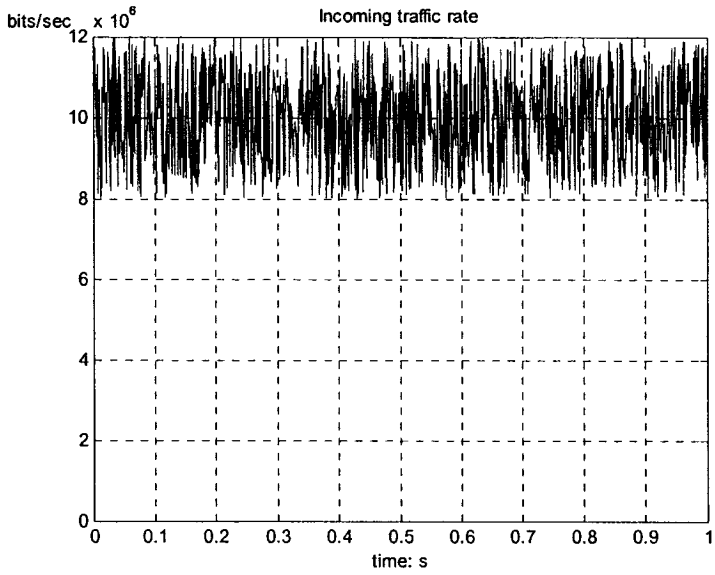

Figure 2-2: Incoming traffic rate

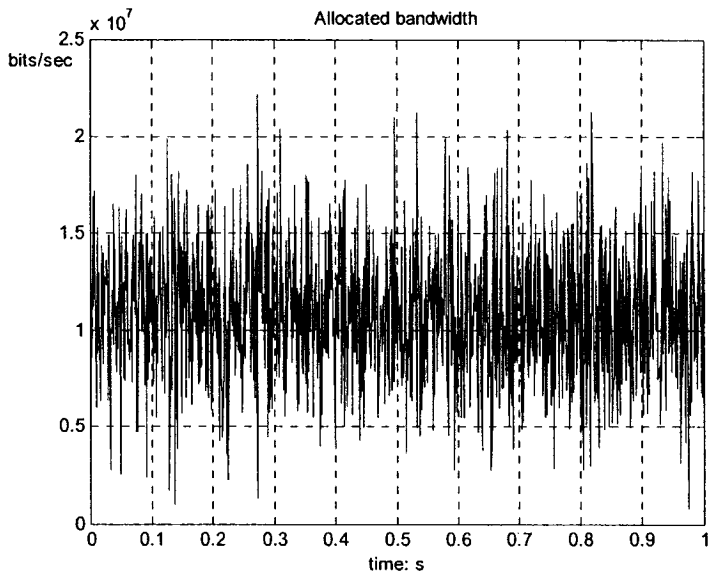

Figure 2-3: Random bandwidth allocated to the traffic

The Simulink model gives the result of queue length shown in Figure 2-4. From this we

notice that the queue length changes at random.

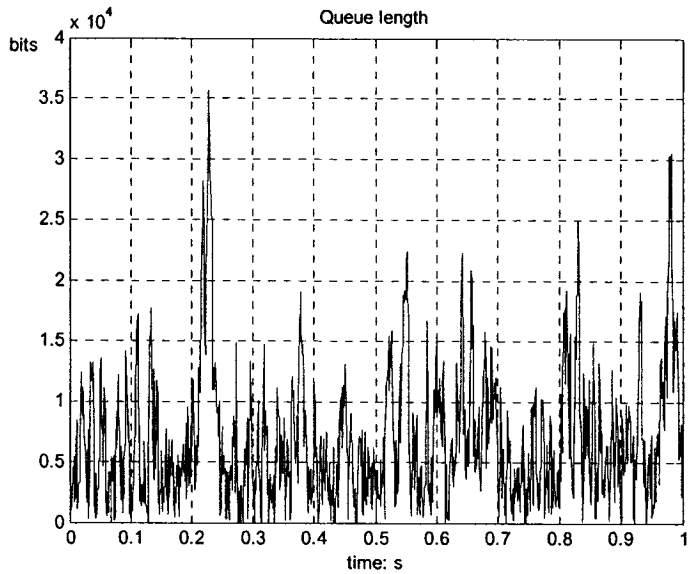

Figure 2-4: Queue length

2.3 Flow fluid model with nonlinear controller

2.3.1 Mathematical model

As shown in Figure 2-3 and Figure 2-4, the queue length and bandwidth are changing randomly. It is not convenient for users or designers to monitor or evaluate the network's performance. In order to archive our control objective, we design the control strategy to attempt to keep a queue buffer length close to a reference value. In this section we will design a simple non-linear congestion controller (the structure is shown in Figure 2-5) to implement this.

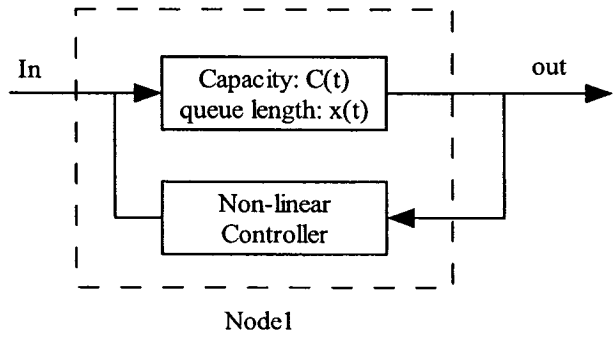

Figure 2-5: Single node with non-linear controller

The control objective is to make the queue length controlled properly. That is, we want the queue length $x(t)$ to be as close to the desired value x (e.g., x^{ref}) as possible. For the differential equation:

$$\dot{x}(t) = \dot{\bar{x}}(t) = \alpha\, \bar{x} = \alpha(x - x^{ref})$$

where $\bar{x} = x - x^{ref}$, x^{ref} is a constant, when $\dot{x}(t) \to 0 \Rightarrow \dot{\bar{x}}(t) \to 0$, that is we will have $x \to x^{ref}$.

Based on the above equations, we design a non-linear controller. The non-linear controller uses the flow rate as input, to calculate the current leftover bandwidth to adjust the incoming traffic. It chooses the capacity $C(t)$ to be allocated to the traffic under the constraint that the incoming traffic rate $\lambda(t)$ is unknown but bounded by k_p.

In mathematical terms we need to choose $C(t)$ so that $\bar{x}(t) \to 0\,(\bar{x} = x - x^{ref})$ under the constraints $C(t) < C_{server}$ and $\lambda(t) \le k_p < C_{server}$. Based on the fluid flow equation, a

feedback linearization and robust adaptive control strategy is proposed in [45] where we select:

$$C(t) = \max\left[0, \min\left\{C_{server}, \rho(t)\frac{1+x(t)}{x(t)}\left[\alpha\bar{x}(t)+k(t)\right]\right\}\right] \tag{2-4}$$

with

$$\rho(t) = \begin{cases} 0 & x(t) \le 0.01 \\ 1.01x(t) - 0.01 & 0.01 \le x(t) \le 1 \\ 1 & x(t) > 1 \end{cases}$$

and

$$\dot{k}(t) = \Pr[\delta\bar{x}(t)] = \begin{cases} \delta\bar{x}(t) & (0 \le k(t) \le k_p) \text{ or } (k(t) = k \text{ and } \bar{x}(t) \le 0) \text{ or } (k(t) = 0 \text{ and } \bar{x}(t) \ge 0) \\ 0 & \text{else} \end{cases}$$

Where k_p is a constant indicating the maximum rate that could be allocated to incoming traffic (e.g. through a connection admission policy), and $\alpha > 0$, $\delta > 0$, are design constants that affect the convergence rate and performance behavior.

For computational reasons the implementation and computation of the above control is performed in discrete time as [16]:

$$k(n+1) = \beta(n)k(n) + \delta(n)\frac{\bar{x}(n)}{\sqrt{1+\bar{x}^2(n)}}$$

where

$$k(n+1) = \begin{cases} k(0) & k(n+1) > C_{server} \text{ or } k(n+1) > k^{\min} \\ k(n+1) & \text{otherwise} \end{cases}$$

where $k(0)$ is chosen as $C_{server}/2$, $0 < k^{\min} << \text{Cserver}$ is a design constant, and

$$\delta(n)=\begin{cases}0 & k(n)>k_p\\ 0 & k(n)\le 0\\ 0.8 & \text{otherwise}\end{cases} \quad \text{and} \quad \beta(n)=\begin{cases}0.9 & k(n)>k_p\\ 1.1 & k(n)\le 0\\ 1 & \text{otherwise}\end{cases}$$

2.3.2 Simulation results

We generate the same incoming traffic as shown Figure 2-2 and we select $\alpha = 2000$. By using the control law in the previous section, we obtain the queue length change as shown in Figure 2-6. As we expected, the queue length increases from time 0, and soon it reaches around the reference value 200Kbits. It is easy to find that at the initial time, the queue length is empty, so it will take time to accumulate bits to reach the designed length.

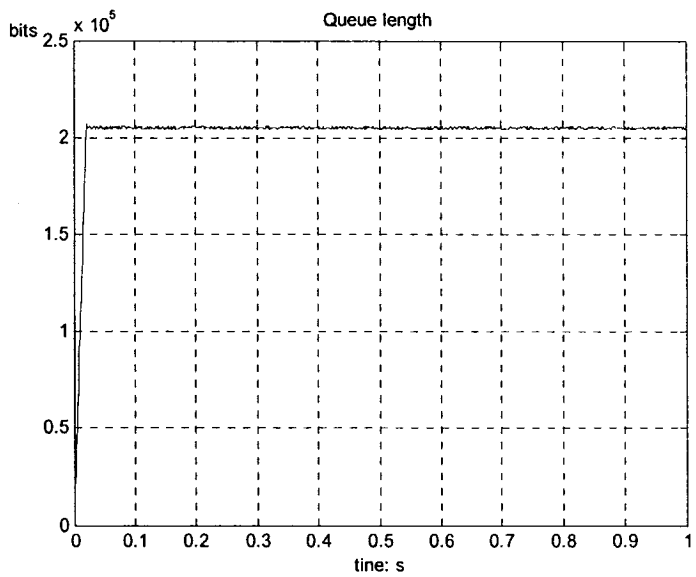

Figure 2-6: Queue length changes under non-linear control

Figure 2-7 shows the bandwidth allocated to the traffic at each time. From Figure 2-6 we

notice that the queue length is well controlled. When queue length reaches its targeted reference, that is $\bar{x}(t) \to 0\ (\bar{x} = x - x^{ref})$, according to (2-3), we will get

$$\dot{x} = -\frac{x(t)}{1+x(t)} * C(t) + \lambda(t) \to 0 \Rightarrow C(t) \approx \lambda(t) \tag{2-5}$$

That means that when the system reaches its equilibrium, the bandwidth allocated to the node approximates the incoming traffic. Comparing Figure 2-2 and Figure 2-7, we notice that the incoming traffic rate approximates the allocated bandwidth, therefore we can conclude that (2-5) is achieved.

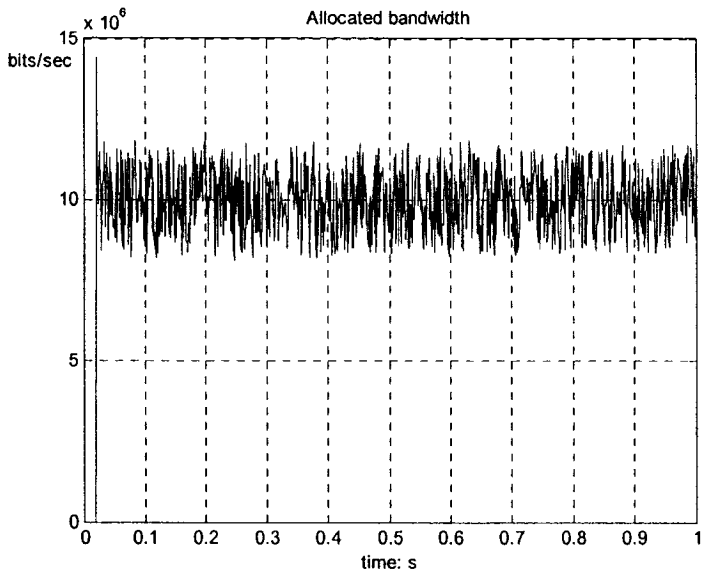

Figure 2-7: Bandwidth allocated to incoming traffic

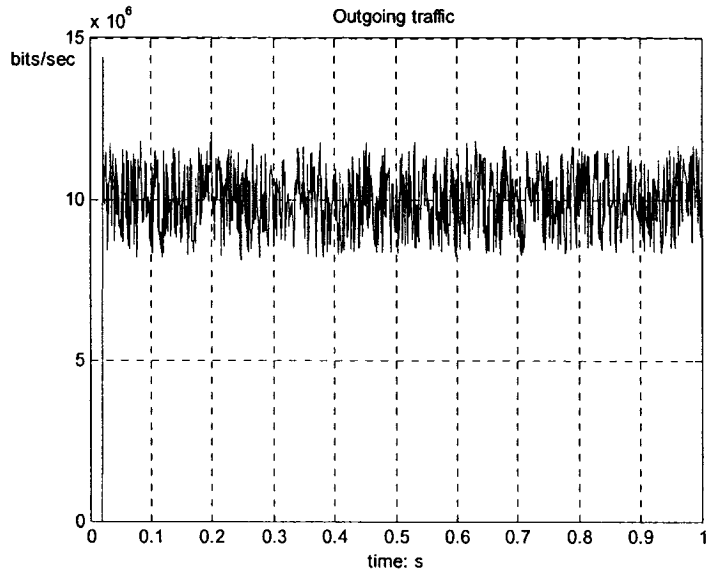

Figure 2-8: Outgoing traffic

Figure 2-8 shows the outgoing traffic at each time. From equations (2-1) and (2-3) we know that

$$f_{out}(t) = -\frac{x(t)}{1+x(t)} * C(t) \tag{2-6}$$

when $x \to x^{ref}$, we can get

$$f_{out}(t) \approx C(t) \tag{2-7}$$

Meanwhile, according to (2-5), we will obtain

$$f_{out}(t) \approx \lambda(t) \tag{2-8}$$

It is easy to understand that when the system achieves its equilibrium, the incoming traffic approximates the outgoing traffic and the queue length remains unchanged.

2.4 Conclusion

In this chapter we introduced a non-linear dynamic model of a network queue (node) based on the fluid flow model and matching the M/M/1 queue behavior at equilibrium. A simple control strategy was introduced to control the queue length. Simulation results and analysis were given to show that the control method is effective.

Chapter 3

A Formal Congestion Algorithm for Single Node

3.1 Description and objectives

In the digital era, more and more networks and data exchanging are needed every second. However, the media of transmission has a limited capacity to afford a large amount of usages. Therefore, the problem of congestion cannot be avoided. How to maximum the usage of bandwidth and avoid the congestion has become an imperative task. Usually there are many users who are sharing the same network link, however, their need for enough bandwidth cannot be granted at all the time. So it is needed to define different priority [15], [16] to each user to make sure that the higher messages get enough bandwidth to pass through.

As shown in Figure 3-1, the incoming traffic is divided into three classes: Premium, Ordinary and Best effort, with its priority from high to low, respectively. We assume that

each incoming traffic flow has a buffer to receive data, and it will share the fixed link capacity C_{server}. Based on the control algorithm discussed in chapter 2, in this chapter we will try to investigate and propose a congestion control scheme to regulate the bandwidth to three different services.

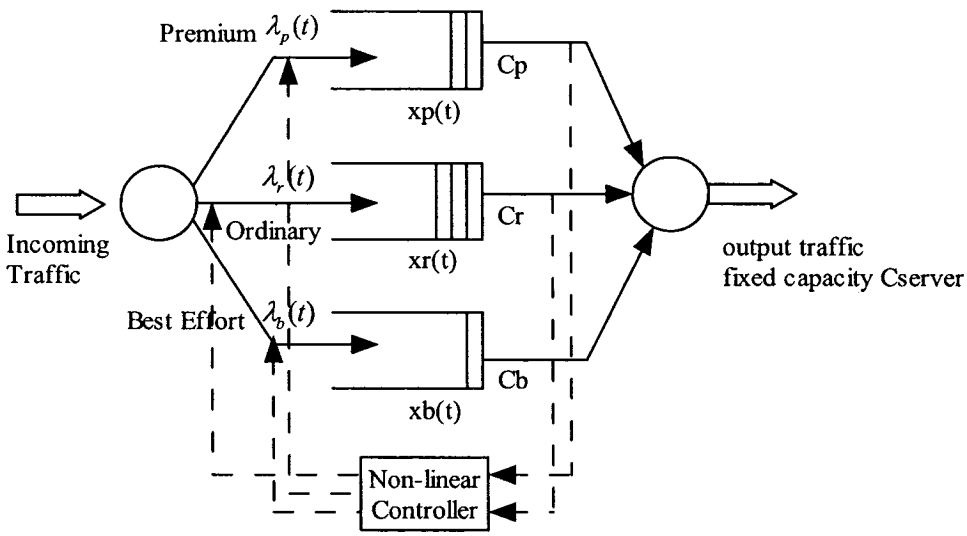

Figure 3-1: Structure of a node with 3 service classes

Our objective is to find an optimal algorithm that can effectively and fairly share the resources among different classes of services. For the structure shown in Figure 3-1, we will make sure that the Premium traffic gets enough bandwidth to pass through, followed by the Ordinary traffic, and finally the Best Effort traffic.

In the following two sections we will present a control strategy to regulate bandwidth allocation among these three classes of services. Some simulation results will be demonstrated as well.

3.2 Proposed control algorithm and model

Based on the method of single queue in chapter 2, for multi services, we need to allocate different bandwidth to each channel (service), thus a non-linear controller is needed to regulate that. For each service, the controller has different control strategy.

For premium traffic service, the proposed approach is to control the length of the premium traffic queue to be always close to a reference value, chosen by the designer so as to indirectly guarantee acceptable bounds for the maximum delay and loss. The capacity for the premium traffic is dynamically allocated, up to the physical server limit, or a given maximum. In this way, the premium traffic is always given resources, up to the allocated maximum (*Cmax*: maximum available or assigned capacity, and *Xmax*: maximum buffer size) to ensure the provision of premium traffic service with known bounds. Whenever this service does not require the use of maximum capacity it offers the excess capacity to the ordinary traffic service.

From the previous chapter and [16], [45], we have

$$\dot{x}_p = -\frac{x_p(t)}{1+x_p(t)} * C_p(t) + \lambda_p(t), \qquad x_p(0) = x_{p0} \tag{3-1}$$

The control objective is to choose the capacity $C_p(t)$ to be allocated to the traffic under the constraint that the incoming traffic rate $\lambda_p(t)$ is unknown but bounded by k_p so that the averaged buffer size $x_p(t)$ is as close to the desired value x (chosen by the user or designer) as possible. In mathematical terms we need to choose $C_p(t)$ so that

$\bar{x}_p(t) \to 0, (\bar{x}_p = x_p - x_p^{ref})$ under the constraints $C_p(t) \le C_{server}$ and $\lambda_p(t) \le k_p < C_{server}$. Based on the method of single queue in chapter 2, the relation between $C_p(t)$ and $x_p(t)$ is given by:

$$C_p(t) = \max\left[0, \min\left\{C_{server}, \rho_p(t)\frac{1+x_p(t)}{x_p(t)}\left[\alpha_p \bar{x}_p(t) + k_p(t)\right]\right\}\right] \tag{3-2}$$

where:

$$\rho_p(t) = \begin{cases} 0 & \text{if } x_p(t) \le 0.01 \\ 1.01x_p(t) - 0.01 & 0.01 \le x_p(t) \le 1 \\ 1 & x_p(t) > 1 \end{cases} \tag{3-3}$$

and

$$\dot{k}_p(t) = \Pr[\delta_p \bar{x}_p(t)] = \begin{cases} \delta_p \bar{x}_p(t) & \text{if } (0 \le k_p(t) \le k_p) \text{or} \\ & (k_p(t) = k_p \text{ and } \bar{x}_p(t) \le 0) \text{or} \\ & (k_p(t) = 0 \text{ and } \bar{x}_p(t) \ge 0) \\ 0 \quad \text{else} \end{cases} \tag{3-4}$$

where K_p is a constant indicating the maximum rate that could be allocated to incoming premium traffic, and $\alpha_p > 0$ and $\delta_p > 0$, are design constants that affect the convergence rate and performance.

For implementation reasons the computation of the above is performed in discrete time as:

$$k_p(n+1) = \beta_p(n)k_p(n) + \delta_p(n)\frac{\bar{x}_p(n)}{\sqrt{1+\bar{x}_p^{\,2}(n)}} \tag{3-5}$$

where

$$k_p(n+1)=\begin{cases}k_p(0) & \text{if } k_p(n+1)>C_{server} \text{ or } k_p(n+1)>k_p{}^{\min}\\ k_p(n+1) & \text{otherwise}\end{cases} \tag{3-6}$$

where $k_p(0)$ is chosen as $\frac{C_{server}}{2}$, $0<k_p{}^{\min}<<C_{server}$ is a design constant, and

$$\delta_p(n)=\begin{cases}0 & k_p(n)>k_p\\ 0 & k_p(n)\le 0\\ 0.8 & \text{otherwise}\end{cases} \quad \text{and} \quad \beta_p(n)=\begin{cases}0.9 & k_p(n)>k_p\\ 1.1 & k_p(n)\le 0\\ 1 & \text{otherwise}\end{cases} \tag{3-7}$$

The controller regulates the flow of ordinary traffic into the network, by monitoring the length of the ordinary traffic queue and the available capacity. If the incoming rate is greater than the maximum left capacity, packets will be hold before flowing into the controller, and they will be transmitted over the next control interval. Under this situation, the algorithm is shown as follows:

$$C_r(t)=\max\left[0,C_{server}-C_p(t)\right] \tag{3-8}$$

Here we design a switching controller (shown in Figure 3-2) to calculate the bandwidth or flow rate for ordinary traffic.

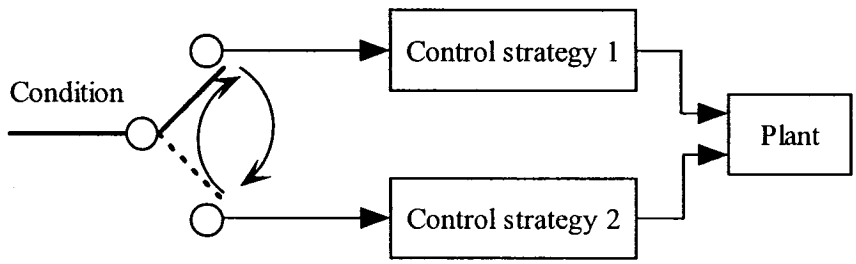

Figure 3-2: Structure of a node with 3 service classes

The length of the ordinary traffic queue is compared with the reference value for implementing this switching controller. The condition is the result after comparing the

leftover bandwidth for ordinary traffic $C_r(t)$ with the incoming traffic rate $\lambda_r(t)$. If the incoming rate is less than the maximum leftover capacity, the premium service algorithm will be applied as:

$$C_r(t) = \max\left[0, \min\left\{C_r, \rho_r(t)\frac{1+x_r(t)}{x_r(t)}\left[\alpha_r \bar{x}_r(t) + k_r(t)\right]\right\}\right] \tag{3-9}$$

Otherwise the length of the ordinary traffic queue is compared with the reference value and using a non-linear control strategy the controlled traffic input rate becomes:

$$\lambda_r(t) = \max\left[0, \min\left\{C_r(t), C_r(t)\frac{1+x_r(t)}{x_r(t)} - \alpha_r \bar{x}_r(t)\right\}\right] \tag{3-10}$$

The best effort service controller calculates the leftover bandwidth after premium and ordinary traffic service by the same switching algorithm as the ordinary traffic, that is

$$C_b(t) = \max\left[0, C_{server} - C_p(t) - C_r(t)\right] \tag{3-11}$$

$$C_b(t) = \max\left[0, \min\left\{C_b, \rho_b(t)\frac{1+x_b(t)}{x_b(t)}\left[\alpha_b \bar{x}_b(t) + k_b(t)\right]\right\}\right] \tag{3-12}$$

$$\lambda_b(t) = \max\left[0, \min\left\{C_b(t), C_b(t)\frac{1+x_b(t)}{x_b(t)} - \alpha_b \bar{x}_b(t)\right\}\right] \tag{3-13}$$

The coefficients in (3-9), (3-10), (3-12) and (3-13) are chosen as the premium service.

3.3 Stability analysis

The control strategy for the premium traffic is described by (3-2) to (3-7), which guarantees that $x_p(t)$ is bounded and converges to the designed value x^{ref}, with error

that depends on the incoming traffic rate $\lambda_p(t)$. $\lambda_p(t)$ is unknown but bounded by k_p and $\lambda_p(t) \le \hat{k}_p < C_{server}$.

Let $\bar{x}_p = x_p - x_p^{ref}$, then $\dot{\bar{x}}_p = \dot{x}_p$, so (3-1) becomes

$$\dot{x}_p = -\frac{x_p(t)}{1+x_p(t)} * C_p(t) + \lambda_p(t), \qquad x_p(0) = x_{p0} \tag{3-14}$$

From (3-2) we know $C_p(t)$ could take the following values over time:

$$C_p(t) = 0 \quad or \quad C_{server} \; or \; \rho_p(t)\frac{1+x_p(t)}{x_p(t)}\left[\alpha_p \bar{x}_p(t) + k_p(t)\right]$$

At the beginning, the buffer is empty, and there is no outgoing traffic. Under this condition, $C_p(t) = 0$, so (3-14) becomes $\dot{x}_p = \lambda_p(t) > 0$, that means the queue length will keep increasing. When $\bar{x}_p \ge 0$, $C_p(t)$ will take the value

$$C_p(t) = \rho_p(t)\frac{1+x_p(t)}{x_p(t)}\left[\alpha_p \bar{x}_p(t) + k_p(t)\right] \tag{3-15}$$

so that $x_p(t)$ can converge to its reference value.

In this situation, according to (3-3), we get $\rho_p(t) = 1$, so (3-14) becomes

$$\dot{x}_p = -\alpha_p \bar{x}_p(t) - [k_p(t) - \lambda_p(t)] \tag{3-16}$$

We propose the following Lyapunov function for stability analysis:

$$V = \frac{\bar{x}_p^{\,2}}{2} + \frac{(k_p - \lambda_p)^2}{2\delta_p} \tag{3-17}$$

According to (3-4) and (3-16) we get

$$\dot{V} = \bar{x}_p \dot{\bar{x}}_p + \frac{(k_p - \lambda_p)(\dot{k}_p - \dot{\lambda}_p)}{\delta_p}$$

$$= -\alpha_p \bar{x}_p{}^2 - (k_p - \lambda_p)\bar{x}_p + (k_p - \lambda_p)\Pr[\bar{x}_p] + \frac{(k_p - \lambda_p)\dot{\lambda}_p}{\delta_p}$$

According to (3-4) and because $\lambda_p(t) \le k_p < C_{server}$, we know that

$$-(k_p - \lambda_p)\bar{x}_p + (k_p - \lambda_p)\Pr[\bar{x}_p] = (k_p - \lambda_p)(-\bar{x}_p + \Pr[\bar{x}_p]) \le 0$$

Therefore

$$\dot{V} \le -\alpha_p \bar{x}_p{}^2 + \sigma \mid \dot{\lambda}_p \mid$$

for some finite nonnegative constant σ.

Because k_p is bounded by (3-4) and $\mid \dot{\lambda}_p \mid$ is bounded from above by a finite constant, so $\bar{x}_p$ is also bounded. There exists $\bar{x}_p$ so that $\dot{V} < 0$, that implies that V decreases as $\bar{x}_p$ increases. This means (3-17) is bounded [16].

When $C_p(t) = C_{server}$, then (3-14) becomes $\dot{x}_p \approx -C_{server} + \lambda_p(t) < 0$, that means the queue length is large that $\bar{x}_p$ will decrease until $C_p(t)$ satisfies (3-12).

So we can find that $\bar{x}_p$ will always be bounded and $x_p(t)$ gets close to x^{ref} with time.

3.4 Simulation method and results

3.4.1 Single incoming traffic

In this section, we investigate the model's performance when the incoming traffic flow (as shown in Figure 3-1) to the node only comes from another one node. First, the method of generating source flow is described, then simulation results are given in the following section.

3.4.1.1 Source model 1

We construct the incoming traffic as described by the following steps. First, we generate random bits stream that is shown in Figure 3-3. Bits stream varies but is bounded to 155Mbits/sec, which is the selected maximum bandwidth.

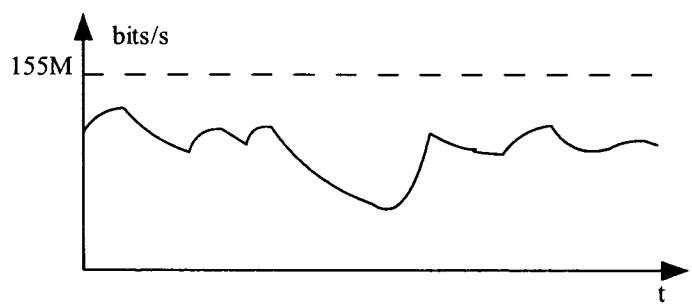

Figure 3-3: Original incoming traffic bits/sec

Second, we sample the incoming bits curve by a fixed sampling time, e.g., 10ms, and collect bits of random time interval to form one group of service data. As shown in Figure 3-3, from 0 to *ts1*, we sample the signal, and pile up the data to service as a

random priority; then do again from *ts1* to *ts2*, *ts2* to *ts3* and so on. The formula is shown below:

$$total\ bits\ in\ one\ service = \int_{0}^{ts1} bits(t)dt \approx Ts \sum_{i=0}^{ts1/Ts} bits(t + i * Ts)$$

where *Ts* is the sample time.

We also define random class-service order and random interval for each order. For instance, in Figure 3-4, bits form 0 to *ts1* represents premium service; those from *ts1* to *ts2* are allocated to ordinary service.

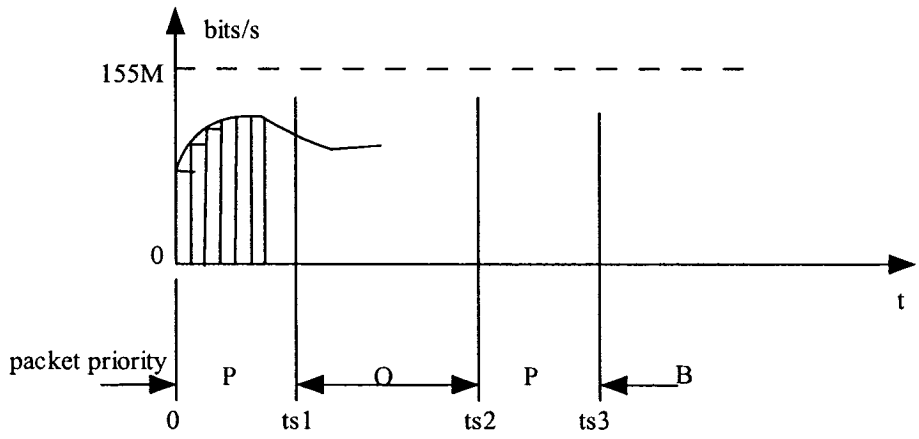

Figure 3-4: Packet content

Finally, those bits will be sent out to the next node (Decision Maker). We choose the bandwidth as 155M bits/sec. Data is sent to the node and served according to different service order. The incoming traffic will be detected and sent to different channel. If its capacity is equal or greater than that outgoing rate, there is no congestion. Otherwise, congestion may occur.

In real environment, for a networked system, the sensor node will detect several different physical signals, for example, image, voice and speed, then send these signals to decision node. We define the image signal as the most important one, then the voice and speed. So in this condition, the image signal represents premium service, it should be served first. When the incoming traffic includes image signal, the system will give the priority to it by getting the enough bandwidth to pass through.

3.4.1.2 Simulation results

Based on the method in the previous section, we construct a group of incoming traffic, and give the proper performance. We choose the link capacity as 155M bits/sec, the control update period $Ts = 1ms$; and the transfer delay form source to node is also $1ms$.

1. Incoming traffic 1

For the premium traffic controller: buffer size as $x_p buffer = 5\text{M}Kbits$, $x_p ref = 200\text{K}bits$, $\alpha_p = 2000$, $k_p = 150 \times 10^6$, $k_p(0) = 80M$, $k_p^{\min} = 800K$.

For the ordinary traffic controller: buffer size as $x_o buffer = 10\text{M}bits$, $x_p ref = 5\text{M}bits$, $\alpha_o = 2000$.

For the best effort traffic controller: buffer size as $x_b buffer = 6Mbits$, $x_b ref = 3Mbits$, $\alpha_b = 2000$.

Based on the above assumptions, we use Matlab and Simulink to simulate the model. We

generate service order (P, O, B) and service length with uniform distribution. Then we generate the incoming bits as random signal with: mean value of incoming rate of premium traffic equals 5Mbits; that of ordinary traffic equals 50Mbits; and that of best effort equals 25Mbits. We choose the mean of the distribution as 50, which means that the service length is 50ms.

Figure 3-5 shows the source bits of 3 classes' services. And Figure 3-6 shows the actual incoming traffic of 3 services to the node, which represent the actual $\lambda(t)$. Because of the sufficient bandwidth, congestion doesn't occur, and we notice that the two figures are identical. In this case, we will use switching control strategy for ordinary and best effort so that both of them will yield enough bandwidth.

Figure 3-7 shows the queue length of each channel. The queue length increases at beginning, and soon it reaches the reference value. Figure 3-8 shows the bandwidth allocated to each service. As the analysis in Chapter 2 shows, we notice that the bandwidth allocated to each service approximates the incoming traffic after the system achieves its equilibrium, which also means that the queue length reaches its designed value. Figure 3-9 shows the total used bandwidth of the node. We notice that there is always enough bandwidth for three incoming traffics.

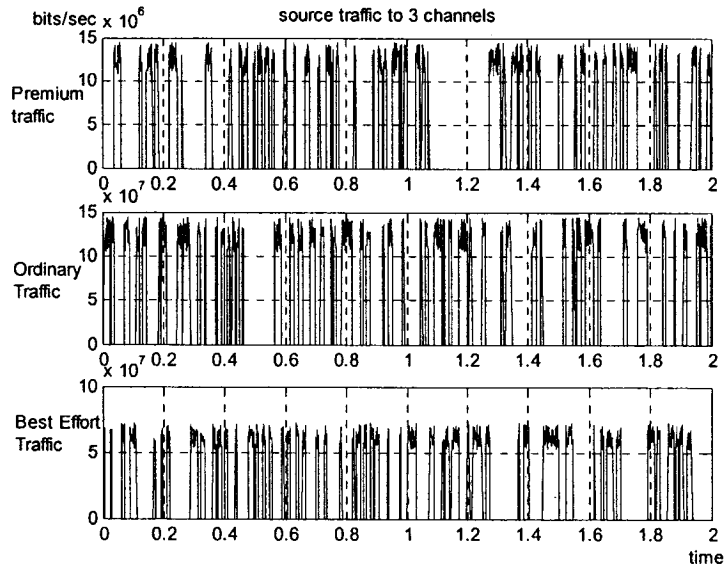

Figure 3-5: Source traffic to 3 channels

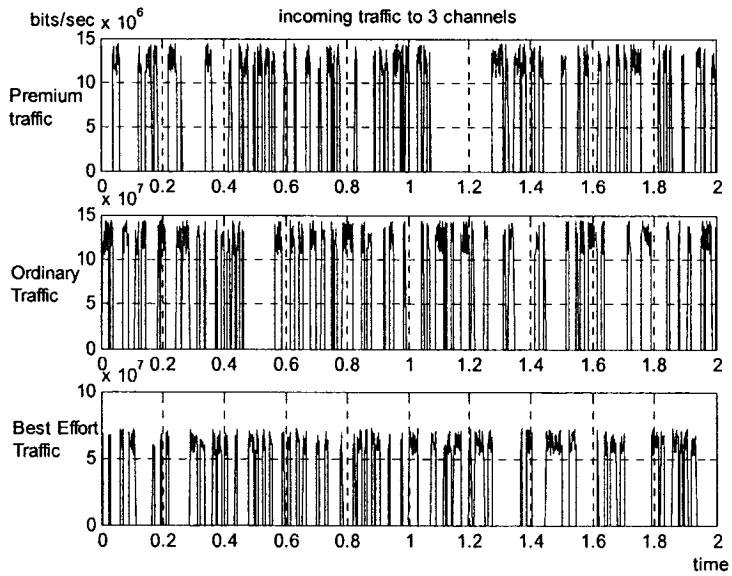

Figure 3-6: Actual incoming traffic for 3 channels

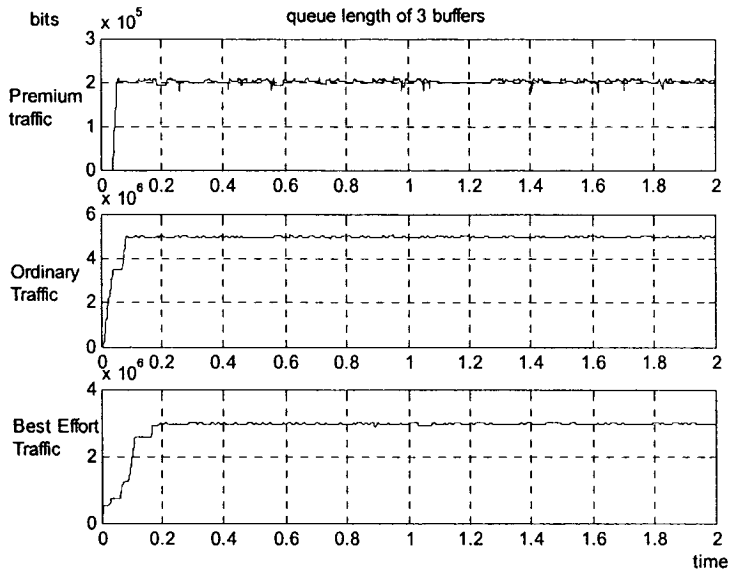

Figure 3-7: Queue length of 3 buffers

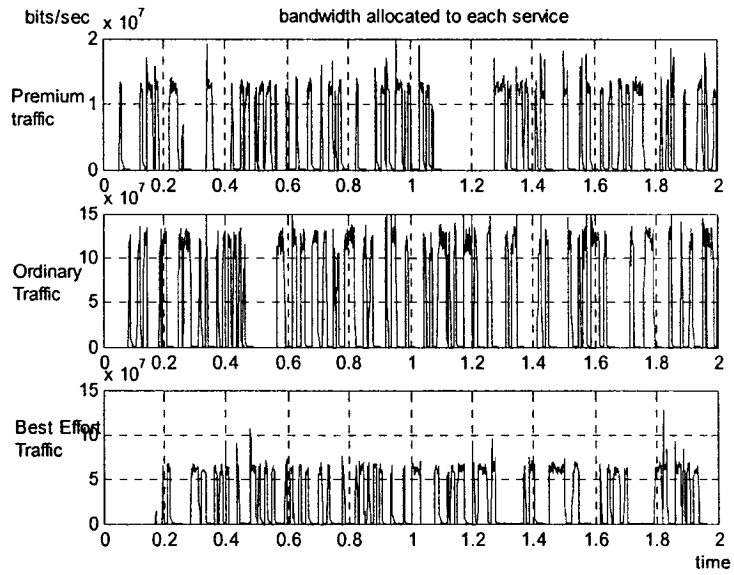

Figure 3-8: Bandwidth allocated to each service

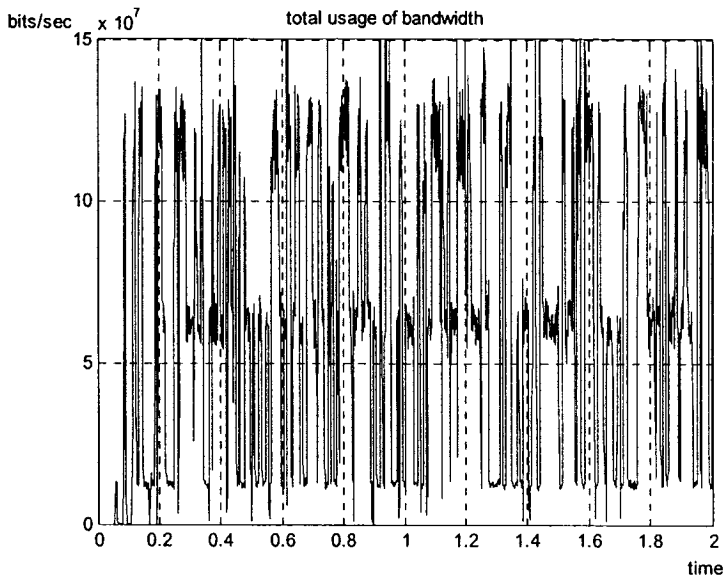

Figure 3-9: Total used bandwidth

3.4.2 Multi incoming traffic

3.4.2.1 Source model 2

If the node receives signal from more than one source (an example is given in Figure 1-2), then the incoming traffic becomes a sum of them. Under this situation, the incoming traffic to node should be continual compared with Figure 3-5.

Based on the algorithm in Section 3.2, we construct three different groups of incoming traffic, and give the proper performance, which represents the model's performance under congestion and dynamic responsibility.

3.4.2.2 Simulation results

2. Incoming traffic 2

We keep all the parameters unchanged and generate the source traffic of each service continually (see Figure 3-10). The mean value of incoming rate of premium traffic equals 10Mbits; that of ordinary traffic equals 100Mbits; and that of best effort equals 40Mbits. Figure 3-11 shows the actual incoming traffic of three services, which represent the $\lambda(t)$. Compared with Figure 3-10, we notice that the source rate of premium and ordinary traffic approximates the incoming rate to node; however, the incoming traffic rate of best effort varies heavily because of limited allocated bandwidth (congestion).

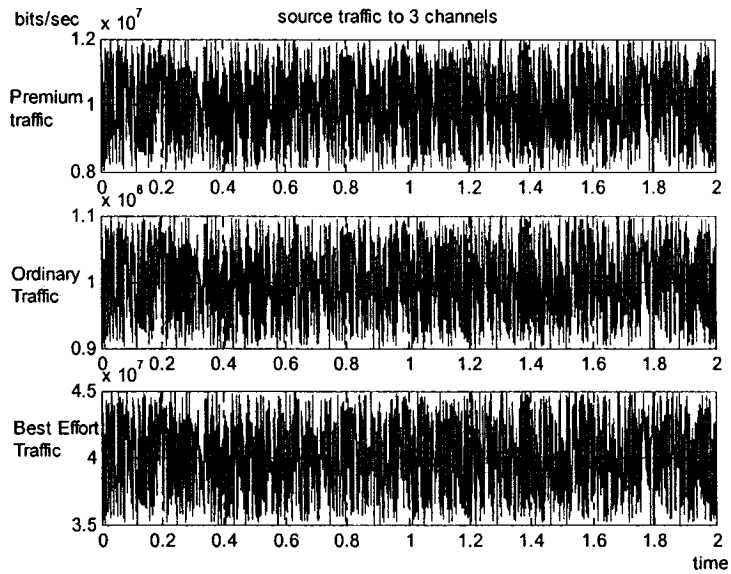

Figure 3-10: Source incoming traffic of 3 channels

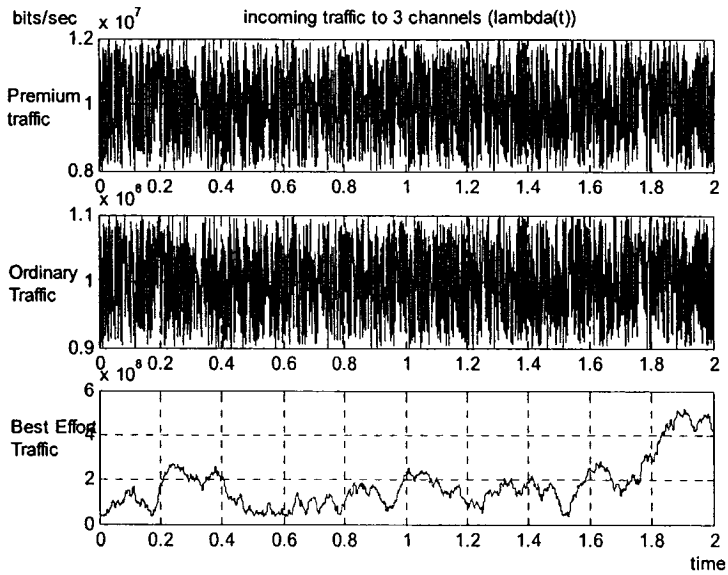

Figure 3-11: Actual incoming traffic for 3 channels because of congestion

Figure 3-12 shows the queue length of each channel. The queue length of premium and ordinary traffic reaches the reference value quickly, however, the queue length of best effort traffic increases much slower than those two because of congestion. Figure 3-13 shows the bandwidth allocated to each service. We notice that the bandwidth allocated to premium and ordinary traffic approximates their incoming traffic. And for best effort traffic, the allocated bandwidth is the leftover after premium and ordinary traffic. The total used bandwidth is shown in Figure 3-14. We notice that it always reaches its maximum value (link capacity).

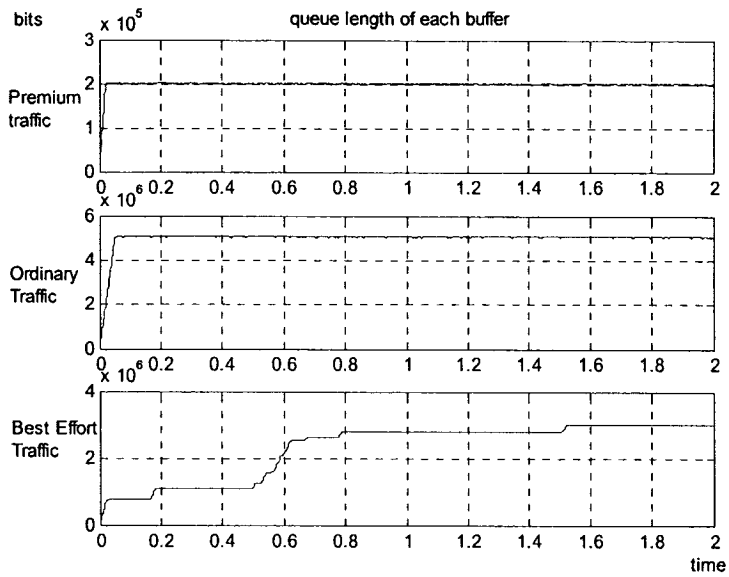

Figure 3-12: Queue length of 3 classes

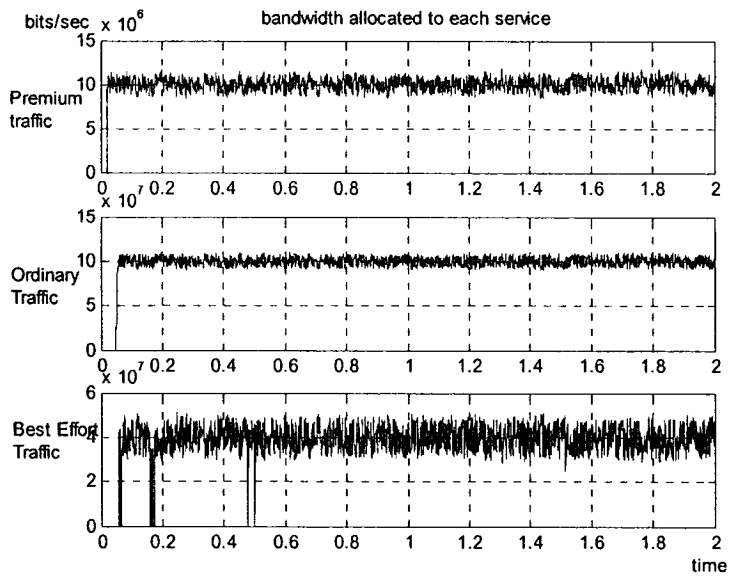

Figure 3-13: Bandwidth allocated to each service

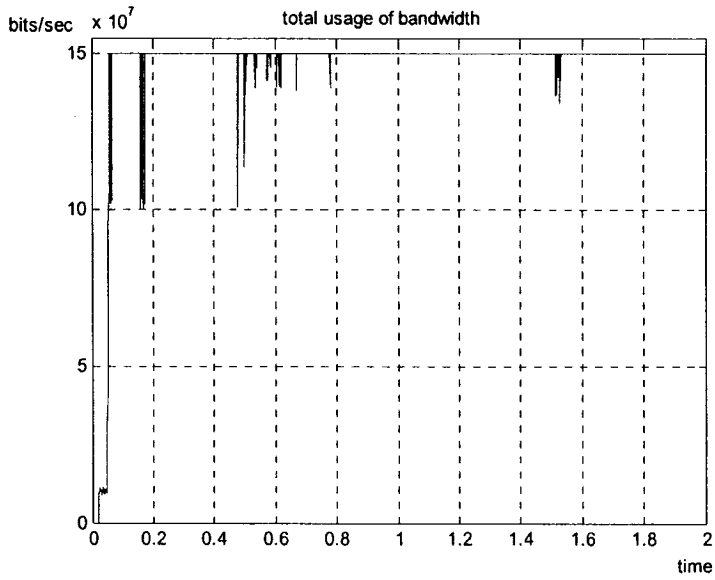

Figure 3-14: Total used bandwidth

3. Incoming traffic 3

As shown in the previous results, the control algorithm achieves a good performance by allocating bandwidth to different services to control the queue length. Now in the following simulations, we want to investigate the dynamic capability of the control strategy.

We keep all the parameters unchanged and we only change the reference value of premium queue length from $200Kbits$ to $100Kbits$ periodically ($T = 0.8s$). The simulation results are as follows.

Figure 3-15 shows an example of the incoming source bits to three channels. Figure 3-16

shows the actual incoming traffic of three services. We notice that the incoming bits of best effort and of ordinary increases in a very short period because of limited allocated bandwidth (congestion).

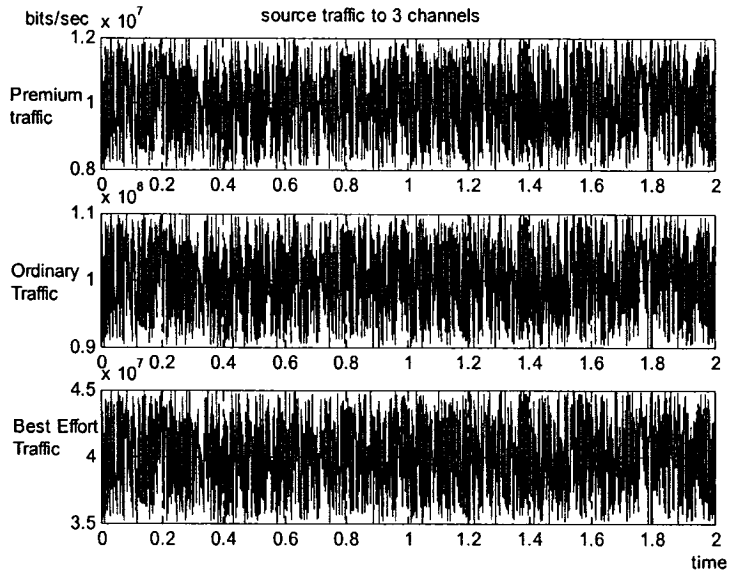

Figure 3-15: Source incoming traffic of 3 channels

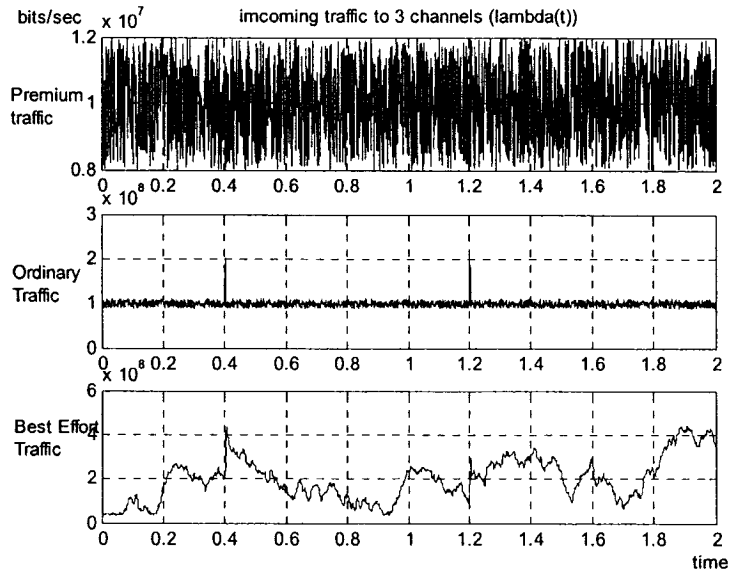

Figure 3-16: Actual incoming traffic for 3 channels because of congestion

Figure 3-17 shows the queue length of each channel. We notice that the queue length of premium changes from 200*Kbits* to 100*Kbits* periodically. That means the queue length is under a good control.

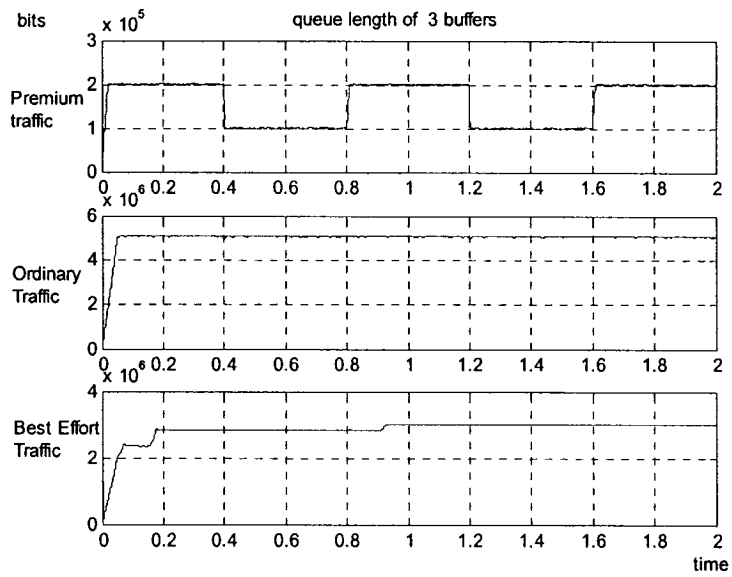

Figure 3-17: Queue length of 3 classes

Figure 3-18 shows the bandwidth allocated to each service. We notice that there are some sharp impulses. This is due to the fact that the reference queue length value of premium traffic changes periodically, so at each time it changes the system will generate an error. However, the queue length reaches its news reference value quickly, which means the system has a good dynamic capability.

The total used bandwidth is shown in Figure 3-19. As shown in Figure 3-14, it always reaches its maximum value (link capacity).

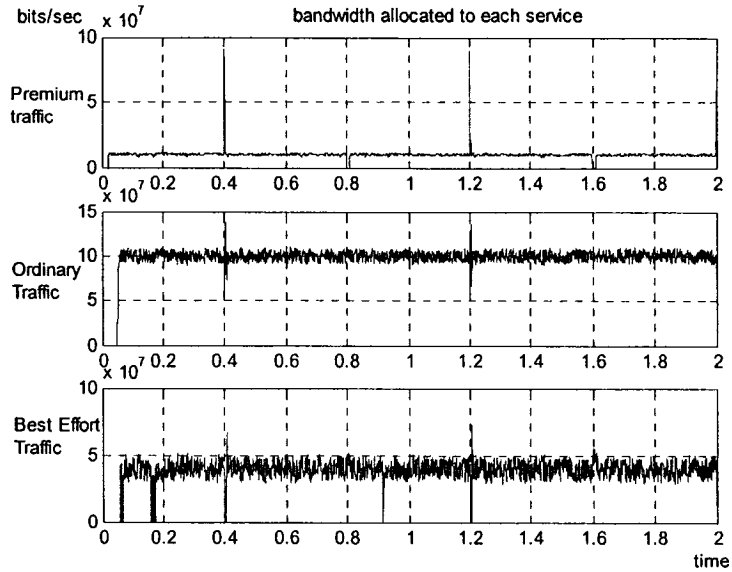

Figure 3-18: Bandwidth allocated to each service

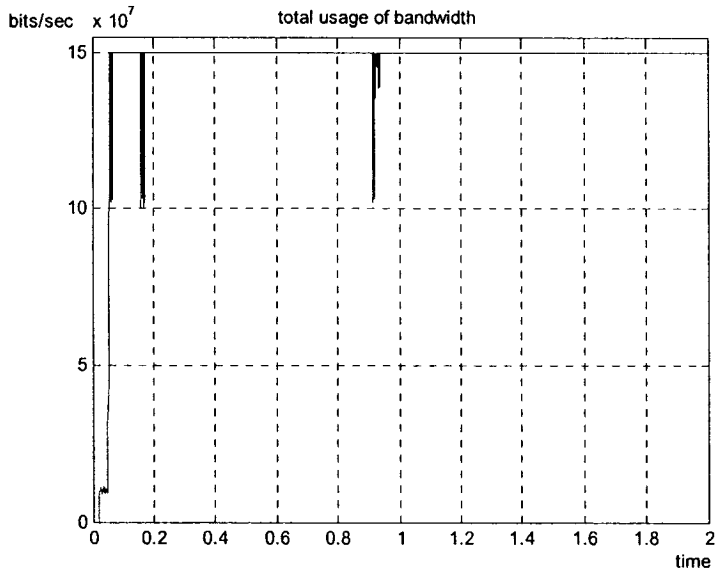

Figure 3-19: Total used bandwidth

From the above simulation results we observe that this control strategy regulates bandwidth allocation among these three classes of services effectively. That guarantees

the high priority services (premium traffic and sometimes ordinary traffic) are served first to get enough bandwidth to pass through. Additionally at the same time, the queue length is controlled approximately.

3.5 Conclusion

In this chapter, we mainly studied the congestion control scheme for regulating the bandwidth to three different services (premium, ordinary and best effort) in one node. We also designed a switching control scheme to switch control algorithm according to the traffic rate. The source incoming traffic from a single input (Figure 3-5) represents insufficient traffic, while that from multi input (Figure 3-10) means sufficient traffic. For both conditions good simulation results were obtained.

Chapter 4

Formal Congestion Algorithms for Multi Nodes

4.1 Description and objectives

In chapter 3, a control strategy is introduced to regulate bandwidth allocation among three classes of services in one node. In practical environment, we usually have a series of nodes in the system. So we should find an effective way to regulate the traffic. This chapter will try to investigate and propose congestion control schemes in a series of nodes (the Sensor- Decision Maker- Actuator channel). Such a structure with feedback information is shown in Figure 4-1.

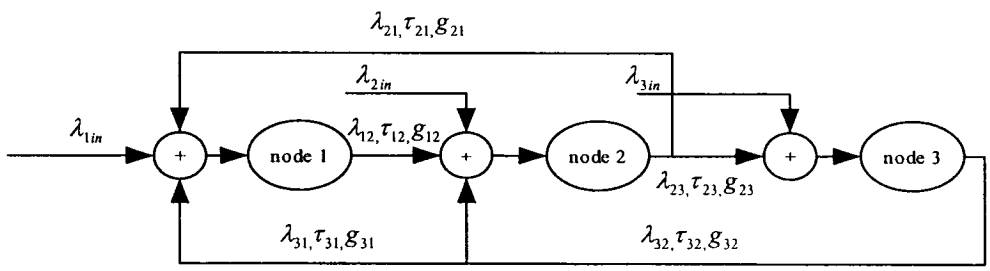

Figure 4-1: Nodes in series-wound with feedback

A simple data flow structure is shown in Figure 4-1: the incoming traffic (here we also define three classes of traffic: premium, ordinary and best effort) streams into the S-DM-A channel, and feedback information (traffic flow) is from Actuator to Sensor and Decision Maker, and from Decision Maker to Sensor. Between every two nodes, we introduce different gains and delays, that is λ defines traffic flow from one node to another or from outside to a node; g represents gain of link between two nodes; and τ is the delay of transmission.

Our objective is to create a model that can effectively and fairly share the resources among different classes of services. The model created for this purpose will attempt to guarantee the premium traffic passes through from node to node, and regulate the incoming rate of ordinary and best effort traffic, to avoid congestion in the whole network.

4.2 Decentralized method

4.2.1 Proposed control algorithm and model

Based on these methods in Section 3.2, we want to design a way to regulate the traffic of a series of nodes. For each node, the control strategy is introduced in Chapter 3. Based on that algorithm, we design a decentralized model for multi nodes' structure. The following Figure 4-2 shows detailed traffic flow structure of Figure 4-1 with $\lambda_{2in} = \lambda_{3in} = 0$.

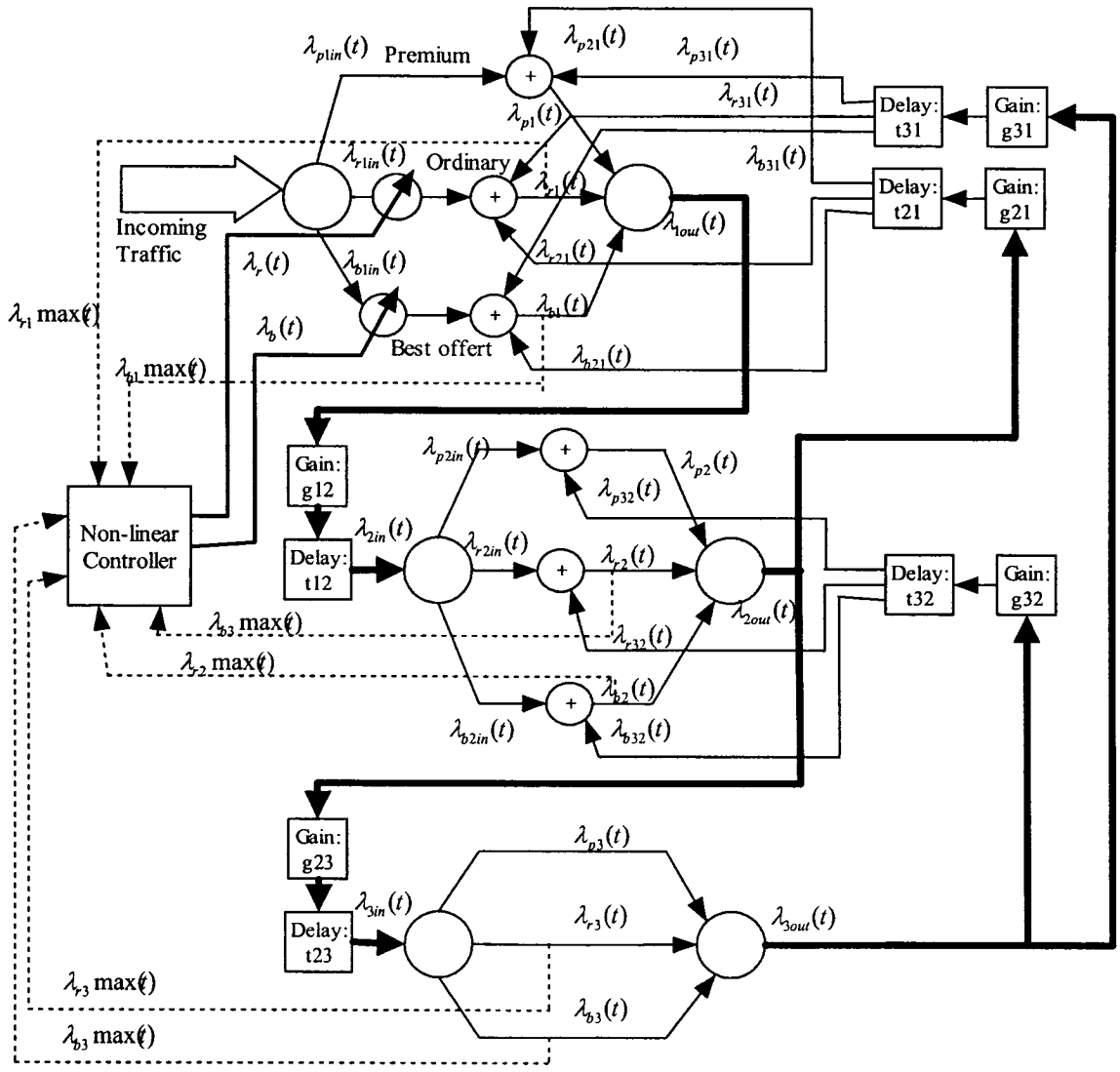

Figure 4-2: Structure of three nodes

First, we assume that the premium traffic is not heavy compared with the whole link capacity (usually 10%). From Figure 4-2 we note that the premium traffic in each node combined with the feedback traffic should be served first. For ordinary traffic, at each node, we calculated the left bandwidth ($C_r(t)$) after premium traffic and feedback traffic, then find available bandwidth to transmit data of the ordinary traffic from one node to

another one. For best effort traffic ($C_b(t)$), method of ordinary traffic is repeated.

Based on the above method, we design a non-linear controller to regulate the incoming rate of ordinary and best effort traffic to avoid congestion in the whole network. For each node, $C_r(t)$ and $C_b(t)$ will be calculated, then compared with the incoming traffic rate, then we will choose the minimum one as the allowed incoming rate of source: $\lambda_r(t)$ and $\lambda_b(t)$.

For premium traffic service, the proposed approach is to control the length of the premium traffic queue to be always close to a reference value, chosen by the designer so as to indirectly guarantee acceptable bounds for the maximum delay and loss. The capacity for the premium traffic is dynamically allocated, up to the physical server limit, or a given maximum constant. In this way, the premium traffic is always given resources, up to the allocated maximum (*Cmax*: maximum available or assigned capacity, and *Xmax*: maximum buffer size) to ensure the provision of premium traffic service with known bounds. Whenever this service does not require the use of maximum capacity it offers the excess capacity to the ordinary traffic service.

For node 1, we have:

$$\dot{x}_{p1} = -\frac{x_{p1}(t)}{1+x_{p1}(t)} * C_{p1}(t) + \lambda_{p1}(t), \qquad x_{p1}(0) = x_{p10} \tag{4-1}$$

$$\lambda_{p1}(t) = \lambda_{p1in}(t) + \lambda_{p21}(t - t_{21}) + \lambda_{p31}(t - t_{31}) \tag{4-2}$$

$$\lambda_{p1out}(t) = \frac{x_{p1}(t)}{1 + x_{p1}(t)} * C_{p1}(t) \tag{4-3}$$

For node 2, we have:

$$\dot{x}_{p2} = -\frac{x_{p2}(t)}{1 + x_{p2}(t)} * C_{p2}(t) + \lambda_{p2}(t), \qquad x_{p2}(0) = x_{p20} \tag{4-4}$$

$$\lambda_{p2in}(t) = \lambda_{p1out}(t - t_{12}) * g_{12} \tag{4-5}$$

$$\lambda_{p2}(t) = \lambda_{p2in}(t) + \lambda_{p32}(t - t_{32}) \tag{4-6}$$

$$\lambda_{p2out}(t) = \frac{x_{p2}(t)}{1 + x_{p2}(t)} * C_{p2}(t) \tag{4-7}$$

$$\lambda_{p21}(t) = \lambda_{p2out}(t - t_{21}) * g_{21} \tag{4-8}$$

For node 3, we have:

$$\dot{x}_{p3} = -\frac{x_{p3}(t)}{1 + x_{p3}(t)} * C_{p3}(t) + \lambda_{p3}(t), \qquad x_{p3}(0) = x_{p30} \tag{4-9}$$

$$\lambda_{p3}(t) = \lambda_{p3in}(t) = \lambda_{p2in}(t - t_{23}) * g_{23} \tag{4-10}$$

$$\lambda_{p3out}(t) = \frac{x_{p3}(t)}{1 + x_{p3}(t)} * C_{p3}(t) \tag{4-11}$$

$$\lambda_{p32}(t) = \lambda_{p3out}(t - t_{32}) * g_{32} \tag{4-12}$$

where:

$x_{pi}(t), (i = 1,2,3)$ represents the state of the queue of premium traffic, given by the ensemble average of the number of bits in the system at time t;

$\lambda_{pi}(t), (i = 1,2,3)$ represents the rate of premium traffic that arrives at the queue;

$\lambda_{pjin}(t), (j = 1,2,3)$ represents the rate of premium traffic that arrives from previous node or source to the next one;

$\lambda_{pij}(t),(i,j=1,2,3)$ represents the feedback rate of premium traffic that arrives from next node to the previous one;

$t_{ij},(i,j=1,2,3)$ represents the delay between transmit channels;

$g_{ij},(i,j=1,2,3)$ represents the gain between transmit channels;

and $C_{pi}(t),(i=1,2,3)$ is the used capacity of queue server for premium traffic.

Note that the above equations are valid for $0 \le x(t) \le x_{buffer\_size}$ and $0 \le C(t) \le C_{server}$.

The control objective in each node is to choose the capacity $C_p(t)$ to be allocated to the traffic under the constraint that the incoming traffic rate $\lambda_p(t)$ is unknown but bounded, so that the averaged buffer size $x_p(t)$ is as close to the desired value x^{ref} (chosen by the operator or designer) as possible. The algorithm is described in Section 2.1.

The ordinary traffic service controller regulates the flow of ordinary traffic into the network, by monitoring the length of the ordinary traffic queue and the available capacity. At each updated time unit, we find the maximum allowed packets rate ($\lambda_{o\max}(t)$) for each node, and then we choose the smallest one as the source-outgoing rate of ordinary service.

For node 1, we have:

$$\lambda_{o1\max}(t) = Cserver - C_{p1}(t) - \lambda_{o21}(t) \tag{4-13}$$

$$\lambda_{o1}(t) = \lambda_{o1in}(t) + \lambda_{o21}(t - t_{21}) + \lambda_{o31}(t - t_{31}) \tag{4-14}$$

$$\lambda_{o1out}(t) = \frac{x_{o1}(t)}{1 + x_{o1}(t)} * C_{r1}(t) \tag{4-15}$$

For node 2, we have:

$$\lambda_{o2\max}(t) = Cserver - C_{p2}(t) - \lambda_{o32}(t - t_{32}) \tag{4-16}$$

$$\lambda_{o2in}(t) = \lambda_{o1out}(t - t_{12}) * g_{12} \tag{4-17}$$

$$\lambda_{o2}(t) = \lambda_{o2in}(t) + \lambda_{o32}(t - t_{32}) \tag{4-18}$$

$$\lambda_{o2out}(t) = \frac{x_{o2}(t)}{1 + x_{o2}(t)} * C_{r2}(t) \tag{4-19}$$

$$\lambda_{o21}(t) = \lambda_{o2out}(t - t_{21}) * g_{21} \tag{4-20}$$

For node 3, we have:

$$\lambda_{o3\max}(t) = Cserver - C_{p3}(t) \tag{4-21}$$

$$\lambda_{o3}(t) = \lambda_{o3in}(t) = \lambda_{o2out}(t - t_{23}) * g_{23} \tag{4-22}$$

$$\lambda_{o3out}(t) = \frac{x_{o3}(t)}{1 + x_{o3}(t)} * C_{r3}(t) \tag{4-23}$$

$$\lambda_{o32}(t) = \lambda_{o3out}(t - t_{32}) * g_{32} \tag{4-24}$$

where:

$x_{oi}(t), (i = 1,2,3)$ represents the state of the queue of ordinary traffic at time t;

$\lambda_{oi}(t), (i = 1,2,3)$ represents the rate of ordinary traffic that arrives at the queue;

$\lambda_{ojin}(t), (j = 1,2,3)$ represents the rate of ordinary traffic that arrives from previous node or source to the next one;

$\lambda_{oij}(t), (i, j = 1,2,3)$ represents the feedback rate of ordinary traffic that arrives from next node to the previous one;

and $C_{ri}(t), (i = 1,2,3)$ represents the used bandwidth of ordinary Traffic, to calculate C_r, we use the same algorithm of Section 2.1.

We regulate the source-outgoing rate of ordinary service using the following formula:

$$\lambda_{oCON}(t) = \min[\lambda_{oin}(t), \quad \lambda_{o1\max}(t), \quad \lambda_{o2\max}(t) \quad \lambda_{o3\max}(t)] \tag{4-25}$$

The best effort service controller calculates the left bandwidth after premium and ordinary traffic service by the same algorithm as the ordinary traffic.

For node 1, we have:

$$\lambda_{b1\max}(t) = Cserver - C_{p1}(t) - C_{r1}(t) - \lambda_{b21}(t) \tag{4-26}$$

$$\lambda_{b1}(t) = \lambda_{b1in}(t) + \lambda_{b21}(t - t_{21}) + \lambda_{b31}(t - t_{31}) \tag{4-27}$$

$$\lambda_{b1out}(t) = \frac{x_{b1}(t)}{1 + x_{b1}(t)} * C_{b1}(t) \tag{4-28}$$

For node 2, we have:

$$\lambda_{b2\max}(t) = Cserver - C_{p2}(t) - C_{r2}(t) - \lambda_{b32}(t - t_{32}) \tag{4-29}$$

$$\lambda_{b2in}(t) = \lambda_{b1out}(t - t_{12}) * g_{12} \tag{4-30}$$

$$\lambda_{b2}(t) = \lambda_{b2in}(t) + \lambda_{b32}(t - t_{32}) \tag{4-31}$$

$$\lambda_{b2out}(t) = \frac{x_{b2}(t)}{1 + x_{b2}(t)} * C_{b2}(t) \tag{4-32}$$

$$\lambda_{b21}(t) = \lambda_{b2out}(t - t_{21}) * g_{21} \tag{4-33}$$

For node 3, we have:

$$\lambda_{b3\max}(t) = Cserver - C_{p3}(t) - C_{r3}(t) \tag{4-34}$$

$$\lambda_{b3}(t) = \lambda_{b3in}(t) = \lambda_{b2out}(t - t_{23}) * g_{23} \tag{4-35}$$

$$\lambda_{b3out}(t) = \frac{x_{b3}(t)}{1 + x_{b3}(t)} * C_{b3}(t) \tag{4-36}$$

$$\lambda_{o32}(t) = \lambda_{o3out}(t - t_{32}) * g_{32} \tag{4-37}$$

where:

$x_{bi}(t),(i=1,2,3)$ represents the state of the queue of best effort traffic at time t;

$\lambda_{bi}(t),(i=1,2,3)$ represents the rate of best effort traffic that arrives at the queue;

$\lambda_{bjin}(t),(j=1,2,3)$ represents the rate of best effort traffic that arrives from previous node or source to the next one;

$\lambda_{bij}(t),(i,j=1,2,3)$ represents the feedback rate of Best Effort traffic that arrives from next node to the previous one;

and $C_{bi}(t)(i=1,2,3)$ represents the used bandwidth of Best Effort Traffic, to calculate C_b, we use the same algorithm of Section 2.1, that is

$$\lambda_{bCON}(t)=\min\left[\lambda_{bin}(t),\quad \lambda_{b1\max}(t),\quad \lambda_{b2\max}(t)\quad \lambda_{b3\max}(t)\right]. \tag{4-38}$$

4.2.2 Stability analysis

Suppose that all the delay units have the same value, τ; and the gain from node 1 to node 2 equals g_1, that from node 2 to node 3 equals g_2, and those of the feedback channels are g_3. For the premium traffic service, from chapter 3 we know that:

$$\begin{aligned} C_p(t) &= \max\left[0,\min\left\{Cserver,\rho_p(t)\frac{1+x_p(t)}{x_p(t)}\left[\alpha_p\bar{x}_p(t)+k_p(t)\right]\right\}\right] \\ &\approx \frac{1+x_p(t)}{x_p(t)}\left[\alpha_p\bar{x}_p(t)+k_p(t)\right] \end{aligned} \tag{4-39}$$

By substituting this term in the dynamic equation, we get:

$$\begin{aligned} \dot{x}_{p1} &= \lambda_{p1in}(t)-\left[\alpha_{p1}\bar{x}_{p1}(t)+k_{p1}(t)\right]+g_3\left[\alpha_{p2}\bar{x}_{p2}(t-2\tau)+k_{p2}(t-2\tau)\right] \\ &\quad +g_3\left[\alpha_{p3}\bar{x}_{p3}(t-3\tau)+k_{p3}(t-3\tau)\right] \end{aligned} \tag{4-40}$$

$$\dot{x}_{p2} = \lambda_{p2in}(t) - \left[\alpha_{p2}\bar{x}_{p2}(t) + k_{p2}(t)\right] + g_1\left[\alpha_{p1}\bar{x}_{p1}(t-\tau) + k_{p1}(t-\tau)\right] + g_3\left[\alpha_{p3}\bar{x}_{p3}(t-2\tau) + k_{p3}(t-2\tau)\right] \quad (4\text{-}41)$$

$$\dot{x}_{p3} = \lambda_{p3in}(t) - \left[\alpha_{p3}\bar{x}_{p3}(t) + k_{p3}(t)\right] + g_2\left[\alpha_{p2}\bar{x}_{p2}(t-\tau) + k_{p2}(t-\tau)\right] \quad (4\text{-}42)$$

For the part including delay, we perform Taylor Series Approximation as follow:

$$u(t-\tau) \xrightarrow{Laplace} u(s)e^{-s\tau} = u(s)(1 - s\tau + \frac{(s\tau)^2}{2!} - ...) \approx u(s)(1 - s\tau) \xrightarrow{Laplace^{-1}} u(t) - \tau\dot{u}(t)$$

Let $\alpha_{p1} = \alpha_{p2} = \alpha_{p3} = \alpha$, thus we get

$$\dot{x}_{p1} = \lambda_{p1in}(t) - \left[\alpha\bar{x}_{p1}(t) + k_{p1}(t)\right] + g_3\left[\alpha\bar{x}_{p2}(t) - 2\alpha\tau\dot{\bar{x}}_{p2}(t) + k_{p2}(t-2\tau)\right] + g_3\left[\alpha\bar{x}_{p3}(t) - 3\alpha\tau\dot{\bar{x}}_{p3}(t) + k_{p3}(t-3\tau)\right] \quad (4\text{-}43)$$

$$\dot{x}_{p2} = \lambda_{p2in}(t) - \left[\alpha\bar{x}_{p2}(t) + k_{p2}(t)\right] + g_1\left[\alpha\bar{x}_{p1}(t) - \alpha\tau\dot{\bar{x}}_{p1}(t) + k_{p1}(t-\tau)\right] + g_3\left[\alpha\bar{x}_{p3}(t) - 2\alpha\tau\dot{\bar{x}}_{p3}(t) + k_{p3}(t-2\tau)\right] \quad (4\text{-}44)$$

$$\dot{x}_{p3} = \lambda_{p3in}(t) - \left[\alpha\bar{x}_{p3}(t) + k_{p3}(t)\right] + g_2\left[\alpha\bar{x}_{p2}(t) - \alpha\tau\dot{\bar{x}}_{p2}(t) + k_{p2}(t-\tau)\right] \quad (4\text{-}45)$$

Because $\bar{x}_{p1} = x_{p1} - x_{ref} \Rightarrow \dot{x}_{p1} = \dot{\bar{x}}_{p1}(t)$, then we get

$$\begin{bmatrix} 1 & 2\alpha g_3\tau & 3\alpha g_3\tau \\ \alpha g_1\tau & 1 & 2\alpha g_3\tau \\ 0 & \alpha g_2\tau & 1 \end{bmatrix}\begin{bmatrix} \dot{\bar{x}}_{p1}(t) \\ \dot{\bar{x}}_{p2}(t) \\ \dot{\bar{x}}_{p3}(t) \end{bmatrix} = \alpha\begin{bmatrix} -1 & g_3 & g_3 \\ g_1 & -1 & g_3 \\ 0 & g_2 & -1 \end{bmatrix}\begin{bmatrix} \bar{x}_{p1} \\ \bar{x}_{p2} \\ \bar{x}_{p3} \end{bmatrix} + \begin{bmatrix} \lambda_{p1in}(t) - k_{p1}(t) + k_{p2}(t-2\tau) + k_{p3}(t-3\tau) \\ \lambda_{p1in}(t) - k_{p1}(t) + k_{p2}(t-2\tau) + k_{p3}(t-3\tau) \\ \lambda_{p1in}(t) - k_{p1}(t) + k_{p2}(t-2\tau) + k_{p3}(t-3\tau) \end{bmatrix} \quad (4\text{-}46)$$

Let

$$A = \begin{bmatrix} 1 & 2\alpha g_3\tau & 3\alpha g_3\tau \\ \alpha g_1\tau & 1 & 2\alpha g_3\tau \\ 0 & \alpha g_2\tau & 1 \end{bmatrix}; B = \alpha\begin{bmatrix} -1 & g_3 & g_3 \\ g_1 & -1 & g_3 \\ 0 & g_2 & -1 \end{bmatrix}$$

$$\dot{\bar{x}}_p(t)=\begin{bmatrix}\dot{\bar{x}}_{p1}(t)\\ \dot{\bar{x}}_{p2}(t)\\ \dot{\bar{x}}_{p3}(t)\end{bmatrix};\bar{x}_p=\begin{bmatrix}\bar{x}_{p1}\\ \bar{x}_{p2}\\ \bar{x}_{p3}\end{bmatrix}$$

$$D=\begin{bmatrix}\lambda_{p1in}(t)-k_{p1}(t)+k_{p2}(t-2\tau)+k_{p3}(t-3\tau)\\ \lambda_{p1in}(t)-k_{p1}(t)+k_{p2}(t-2\tau)+k_{p3}(t-3\tau)\\ \lambda_{p1in}(t)-k_{p1}(t)+k_{p2}(t-2\tau)+k_{p3}(t-3\tau)\end{bmatrix}$$

Thus we obtain:

$$\dot{\bar{x}}_p(t)=A^{-1}B\bar{x}_p+A^{-1}D \tag{4-47}$$

If we want the queue length status $\bar{x}_p$ to be stable, we need the matrix $A^{-1}B$ to be stable, that is

$$P=A^{-1}B=\frac{A^*B}{|A|}=$$

$$\frac{\begin{bmatrix}2\alpha^2g_2g_3\tau^2-1+\alpha g_1g_3\tau(3\alpha g_2\tau-2) & (1-2\alpha^2g_2g_3\tau^2)g_3-\alpha g_3\tau(3\alpha g_2\tau-2)+\alpha g_2g_3\tau(4\alpha g_3\tau-3) & (1-2\alpha^2g_2g_3\tau^2)g_3+\alpha g_3{}^2\tau(3\alpha g_2\tau-2)-\alpha g_3\tau(4\alpha g_3\tau-3)\\ g_1+\alpha g_1\tau & -\alpha g_1g_3\tau-1+\alpha g_2g_3\tau(3\alpha g_1\tau-2) & g_3-\alpha g_1g_3\tau-\alpha g_3\tau(3\alpha g_1\tau-2)\\ -\alpha g_1g_2\tau-\alpha^2g_1g_2\tau^2 & \alpha^2g_1g_2g_3\tau^2+\alpha g_2\tau+(1-2\alpha^2g_1g_3\tau^2)g_2 & \alpha^2g_1g_2g_3\tau^2-\alpha g_2g_3\tau-1+2\alpha^2g_1g_3\tau^2\end{bmatrix}}{1-2\alpha^2g_2g_3\tau^2-2\alpha^2g_1g_3\tau^2+3\alpha^3g_1g_2g_3\tau^3} \tag{4-48}$$

According to the Routh-Hurwitz stability criterion, we need that

$$eq1=-(p_{11}+p_{22}+p_{33})>0 \tag{4-49}$$

$$eq2 = p_{11}p_{22} + p_{11}p_{33} + p_{22}p_{33} - p_{13}p_{31} - p_{32}p_{23} - p_{12}p_{21} > 0 \qquad (4\text{-}50)$$

$$eq3 = p_{11}p_{23}p_{32} + p_{22}p_{13}p_{31} + p_{33}p_{12}p_{21} - p_{11}p_{22}p_{33} > 0 \qquad (4\text{-}51)$$

$$eq4 = eq1 * eq2 - eq3 > 0 \qquad (4\text{-}52)$$

where $p_{ij}(i, j = 1,2,3)$ is the element in the matrix P. (The Proof of the derivation is shown in Appendix A).

Based on the above criteria, we can find the relation and constraints of the four variables. For example, we choose $g_1 = g_2 = g_3 = 0.1$, then we calculate τ according to criteria (4-49) to (4-52), to get the $\tau_{\max} = 8ms$.

4.2.3 Simulation method and result

We construct the data as the method in Section 3.4.2. We do the following simulations. We choose the total link capacity as 155M bits/sec, the control update period $Ts = 1ms$; transfer delay is also 1ms. Let $t_{12} = t_{21} = t_{23} = t_{32} = \tau$, $g_{12} = g_1$, $g_{23} = g_2$, $g_{21} = g_{32} = g_{31} = g_3$, $\lambda_{2in} = \lambda_{3in} = 0$.

4. Simulation strategy 1

For Nodes 1, 2 and 3, we choose:

For the premium traffic controller: buffer size as $x_p buffer = 5\text{M}bits$, $x_p ref = 200\text{K}bits$, $\alpha_p = 500$, $\hat{k}_p = 150 \times 10^6$, $k_p(0) = 80M$, $k_p^{\min} = 800K$. For the ordinary traffic controller: buffer size as $x_o buffer = 10\text{M}bits$, $x_o ref = 5\text{M}bits$, $\alpha_o = 500$. For the best

effort traffic controller: buffer size as $x_b buffer = 6Mbits$, $x_b ref = 3Mbits$, $\alpha_b = 500$. We choose the delay $\tau = 0.001s$ and the gain $g_1 = g_2 = 0.9$, $g_2 = 0.1$.

Based on the above assumptions, we use Simulink to simulate this model. We generate the source traffic of each service continually (see Figure 4-3). The mean value of λ_{1in} (incoming rate) of premium traffic equals 10Mbits; that of ordinary traffic equals 100Mbits; and that of best effort equals 40Mbits.

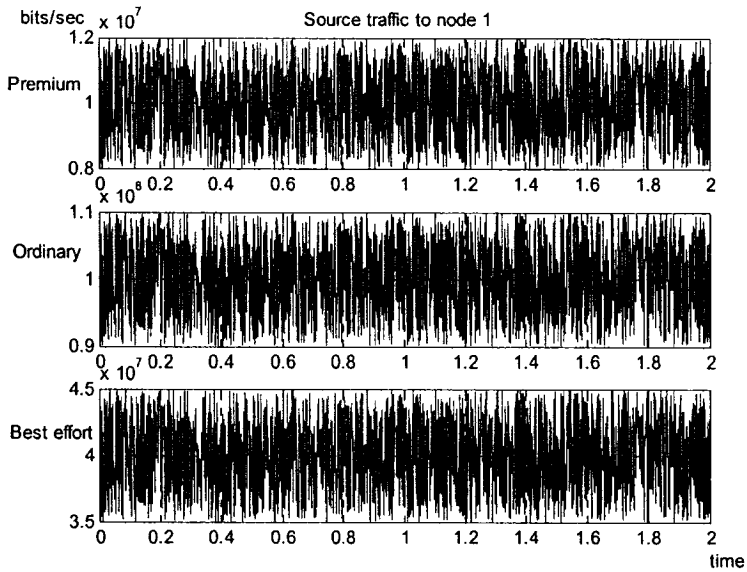

Figure 4-3: Source traffic to node 1

At each node, we calculate the maximum allowed rate of ordinary traffic and best effort traffic according to algorithm in Section 4.2, which is shown in Figures 4-4 and 4-6. We note that the maximum allowed rate is the whole capacity at the beginning, because during that period packets of premium traffic are accumulating in node 1 and there is no premium traffic in node 2 and 3. After several updated periods, the maximum allowed

rate reaches stable value because the premium traffic has achieved its equilibrium (Shown in Figures 4-5 and 4-7).

In Figures 4-5 and 4-7, we compare the source traffic of ordinary and best effort service and the actual controlled rate (which is the smallest one of Figures 4-4and 4-6), and find that there is enough bandwidth for ordinary service, so the left traffic is zero (Figure 4-5); while for best effort traffic, the allowed rate is far slower than the incoming rate, so the leftover traffic increases because of congestion (Figure 4-7).

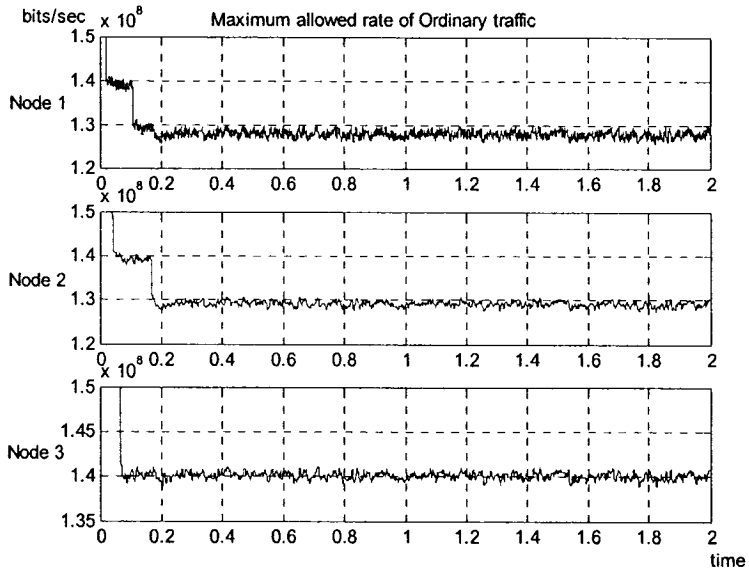

Figure 4-4: Maximum allowed rate of ordinary traffic

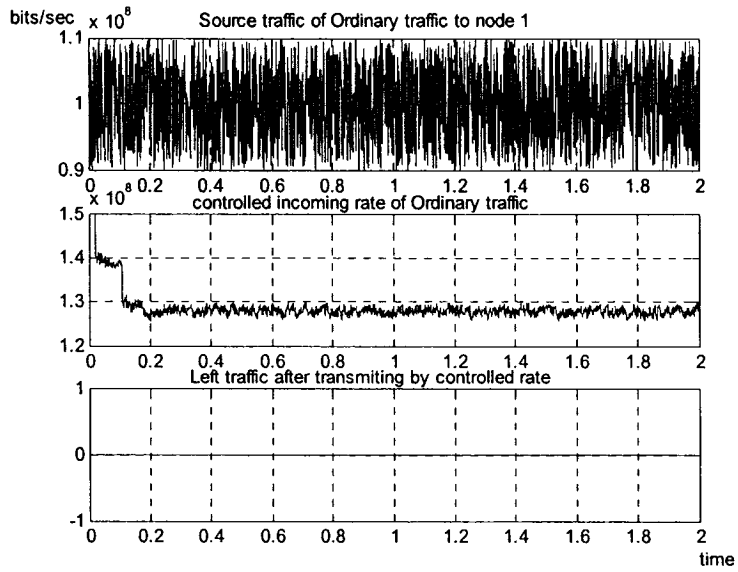

Figure 4-5: Source and actual rate and leftover traffic of ordinary service

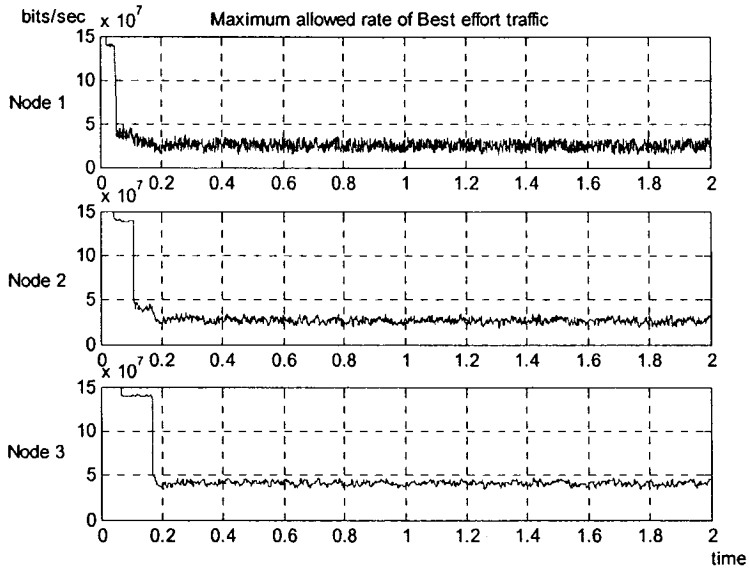

Figure 4-6: Maximum allowed rate of best effort traffic

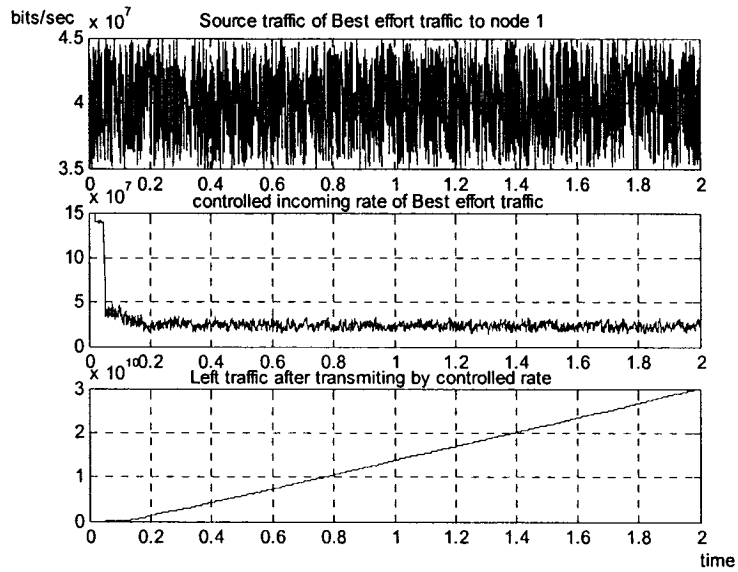

Figure 4-7: Source and actual rate and leftover traffic of best effort service

Figure 4-8 shows the jointed incoming traffic to node 1. For premium service and ordinary service, they are composed of the source traffic, feedback traffic from node 2 and node 3; for best effort service, it represents the traffic under control. For node 2 (Figure 4-9), it receives the outgoing traffic of node 1 and feedback traffic from node 3. However, node 3 (Figure 4-10) only receives the traffic from node 2. Comparing the following three figures with Figure 4-3, we notice that for premium and ordinary traffic the actual incoming rates are greater than the source rate, because they include the feedback traffic; however, the rate of best effort service is less than the source rate because the maximum allowed rate is limited.

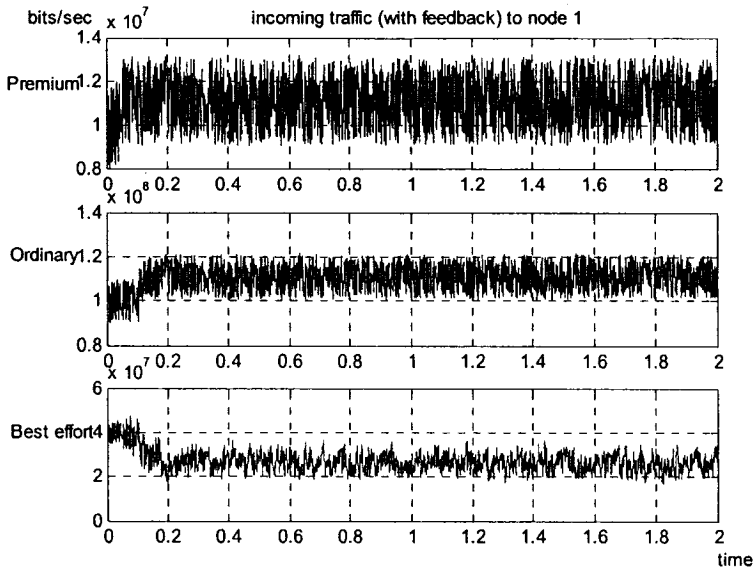

Figure 4-8: Actual incoming traffic to node 1

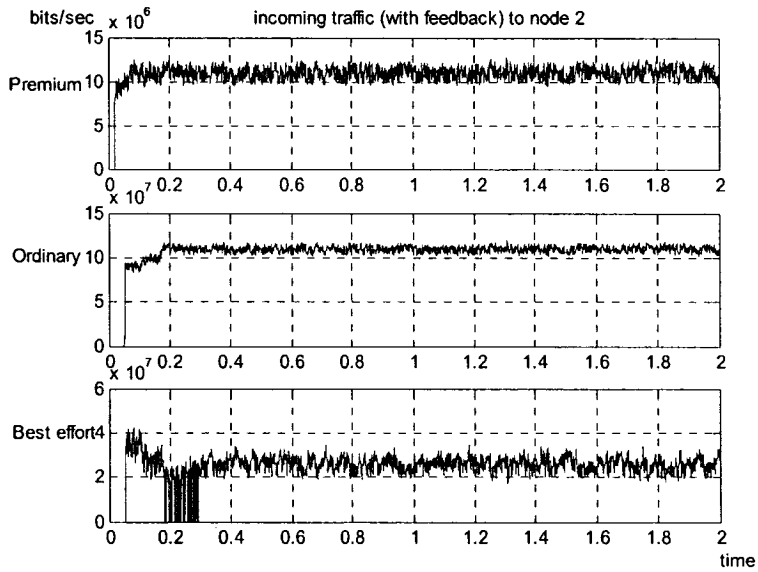

Figure 4-9: Actual incoming traffic to node 2

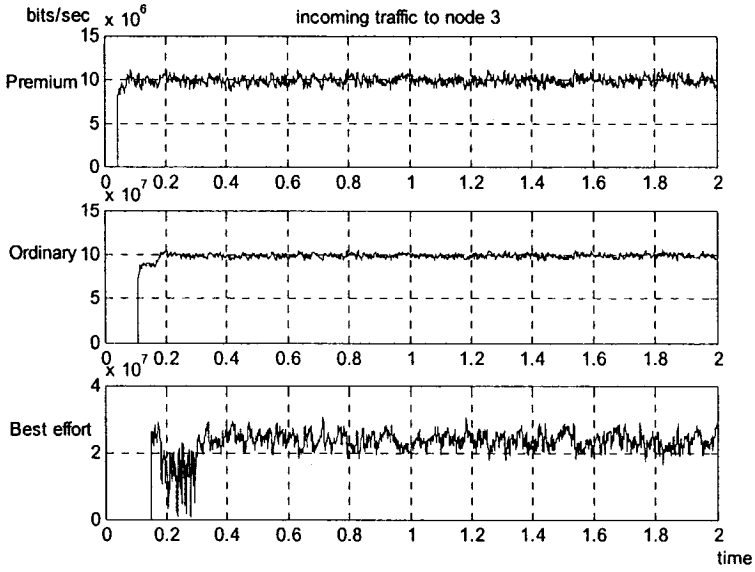

Figure 4-10: Actual incoming traffic to node 3

The queue length of each service in each node is shown in Figures 4-11, 4-12 and 4-13. We note that all queues have reached the preset values, so they are well controlled.

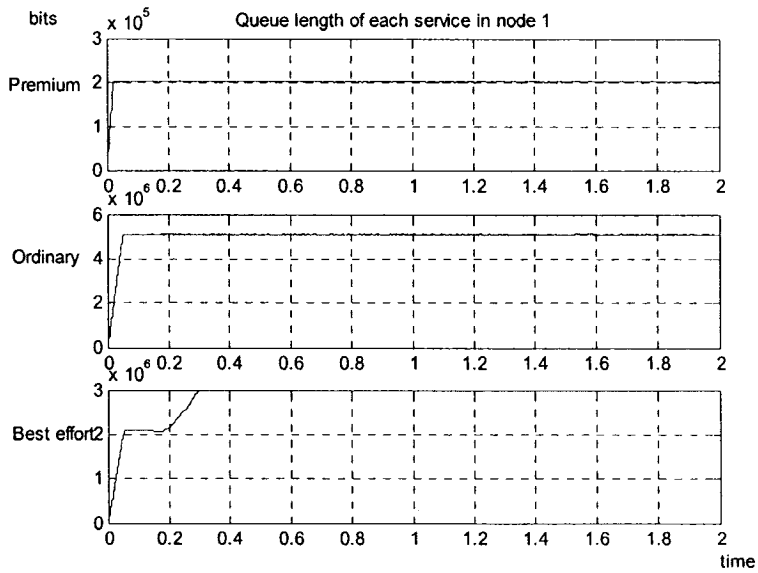

Figure 4-11: Queue length of each service in node 1

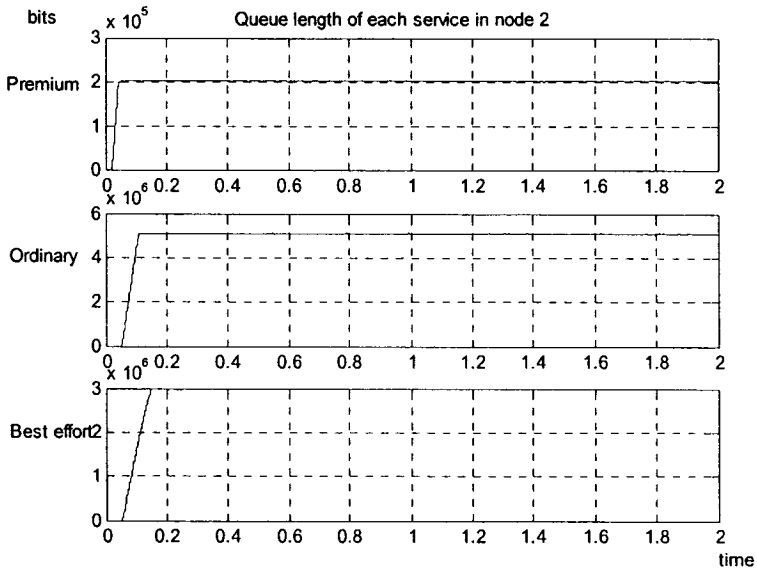

Figure 4-12: Queue length of each service in node 2

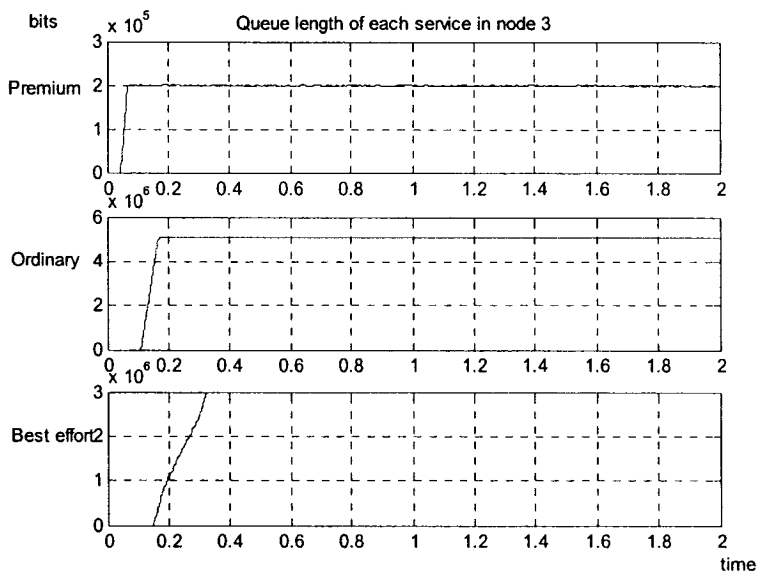

Figure 4-13: Queue length of each service in node 3

Figures 4-14, 4-15 and 4-16 show the bandwidth allocated to each service in each node.

According to the formula in section 2, we have

$$\dot{x} = -\frac{x(t)}{1+x(t)} * C(t) + \lambda(t), \qquad x(0) = x_0$$

We know that if the queue length could reach the reference value, we have

$$\dot{x} = 0 \quad \text{and}$$

$$\frac{x(t)}{1+x(t)} * C(t) = \lambda(t) \Rightarrow C(t) \approx \lambda(t)$$

It requires the incoming rate approximates the used capacity, which can be seen by comparing Figures 4-8, 4-9, 4-10, 4-14, 4-15 and 4-16.

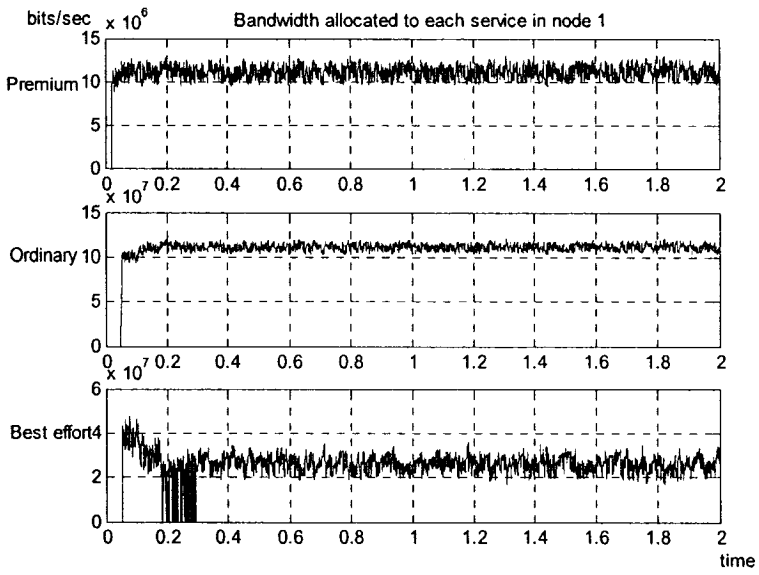

Figure 4-14: Bandwidth allocated to each service in node 1

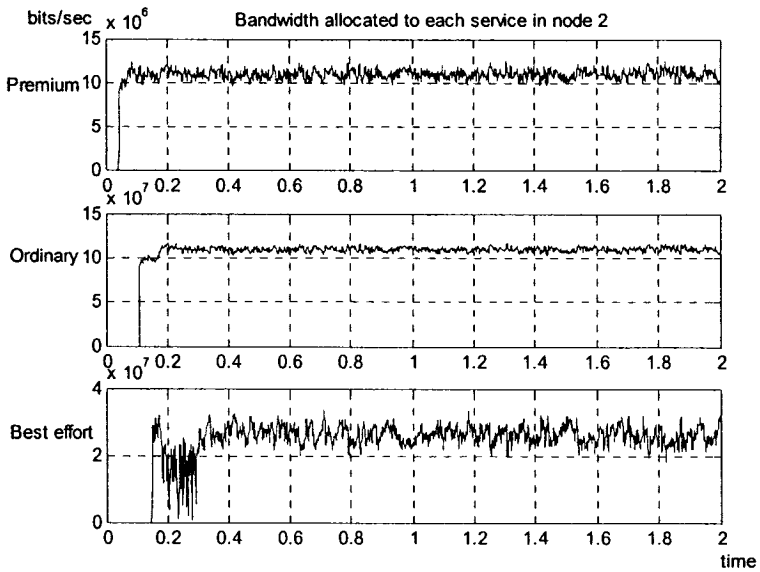

Figure 4-15: Bandwidth allocated to each service in node 2

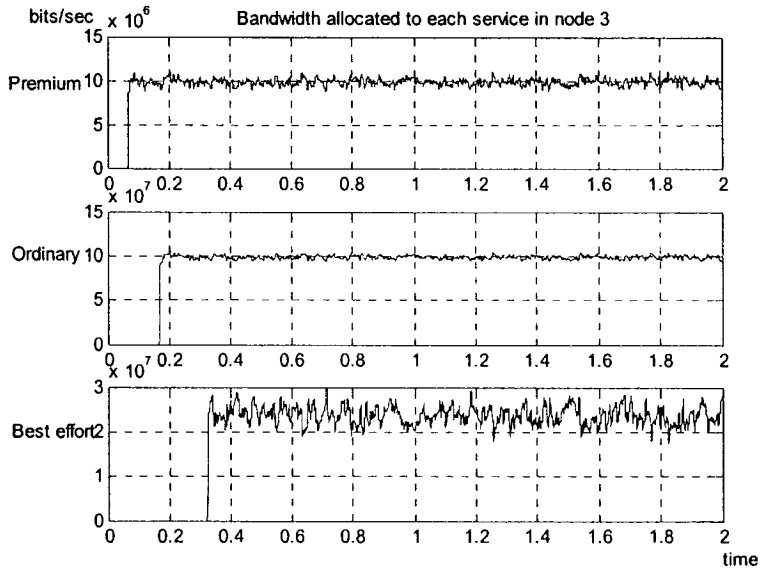

Figure 4-16: Bandwidth allocated to each service in node 3

Figure 4-17 shows the total used bandwidth of each node. We notice that for nodes 1 and 2, the used bandwidth approximates the link capacity, which means the control algorithm

works well. For node 3, there is no sufficient traffic, especially no feedback from other nodes, so the used bandwidth is far less than the link capacity.

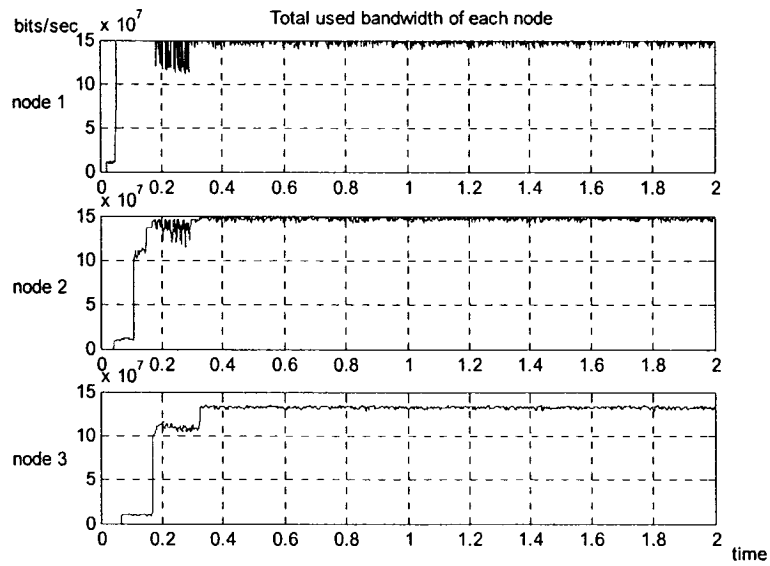

Figure 4-17: Total used bandwidth of each node

5. Simulation strategy 2

We keep all the parameters unchanged, we change $g_1 = g_2 = g_3 = 0.5$ and $x_p ref = 500K bits$. Then we generate the source traffic of each service continually as shown in Figure 4-3.

At each node, we calculate the maximum allowed rate of ordinary traffic and best effort traffic which is shown in Figures 4-18 and 4-20. In Figures 4-19 and 4-21, we compare the source traffic of ordinary and best effort service with the actual controlled rate (which is the smallest one of Figures 4-18 and 4-20), we note that although the incoming traffic

is heavy, but the maximum allowed rate is limited. There is insufficient bandwidth for both ordinary and best effort service, so congestion occurs (shown in Figures 4-19 and 4-21), which means incoming packets are hold before flowing into node 1.

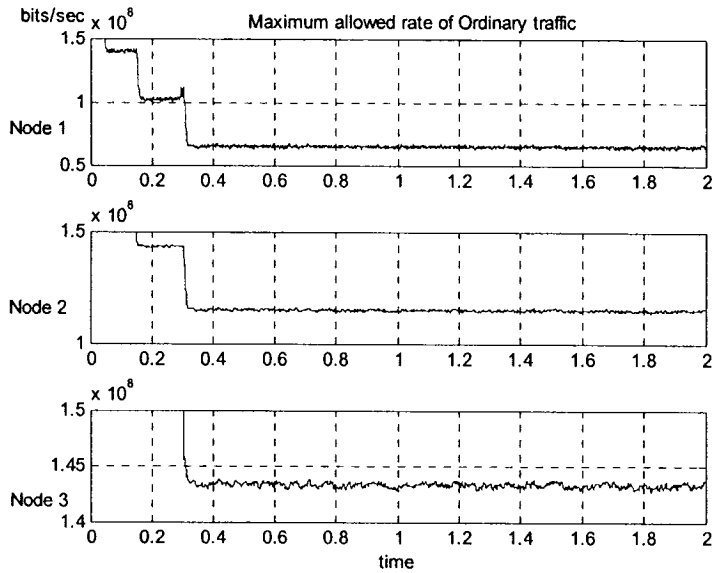

Figure 4-18: Maximum allowed rate of ordinary traffic

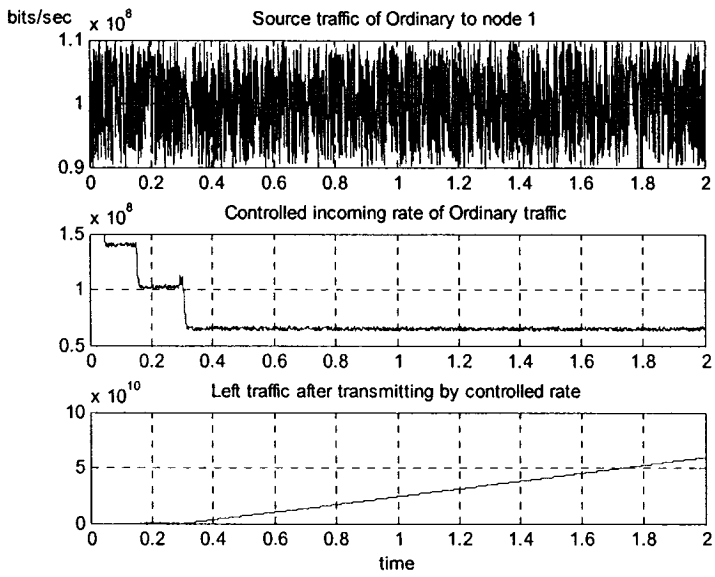

Figure 4-19: Source and actual rate and leftover traffic of ordinary service

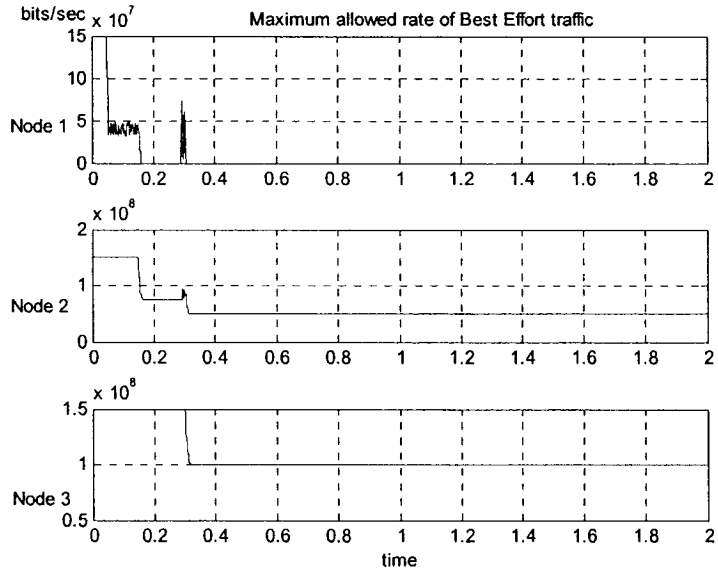

Figure 4-20: Maximum allowed rate of best effort traffic

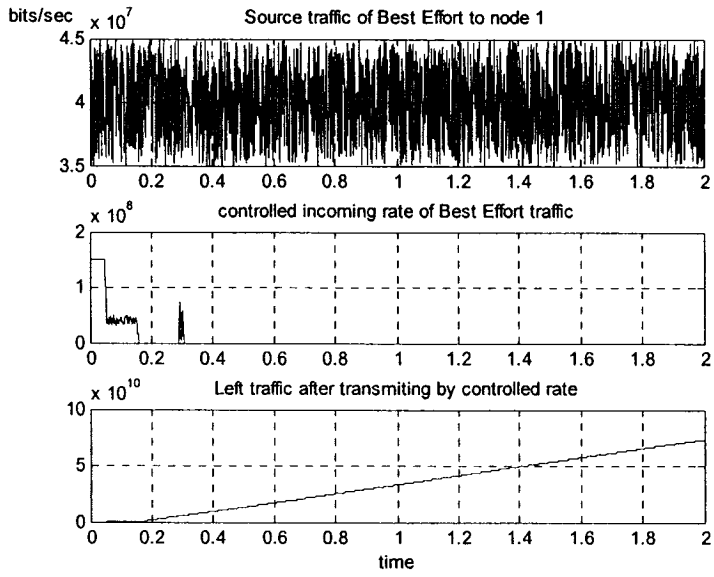

Figure 4-21: Source and actual rate and leftover traffic of best effort service

Figures 4-22, 4-23 and 4-24 show the jointed incoming traffic to nodes 1, 2 and 3, respectively. For premium service, it is composed of the source traffic and feedback traffic from nodes 2 and 3; for ordinary service and best effort service, it goes under controlled rate. We note that for best effort service, there is limited traffic in nodes 1 and 2; however, it becomes zero in node 3 because there is no output traffic from node 2. The ordinary rate is low at the beginning; then increases after feedback traffic are available. It matches the controlled rate shown in Figure 4-19. For node 2 (Figure 4-23), it receives the outgoing traffic of node 1 and feedback traffic from node 3. However, node 3 (Figure 4-24) only accepts the traffic from node 2.

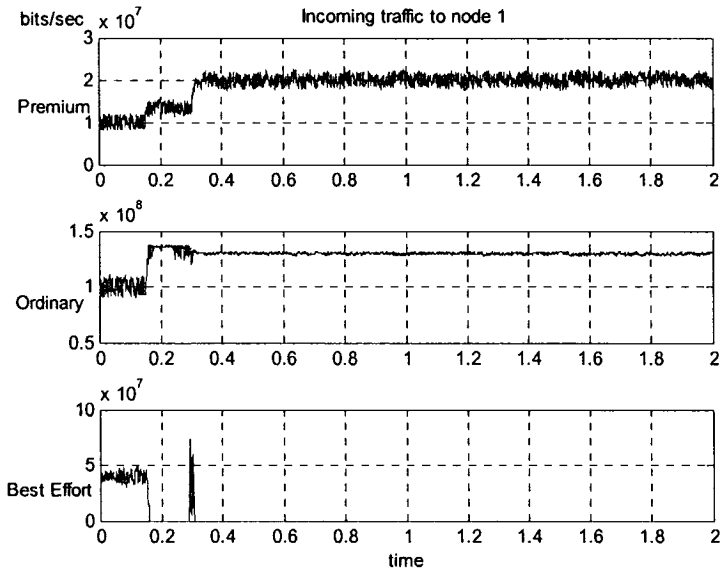

Figure 4-22: Actual incoming traffic to node 1

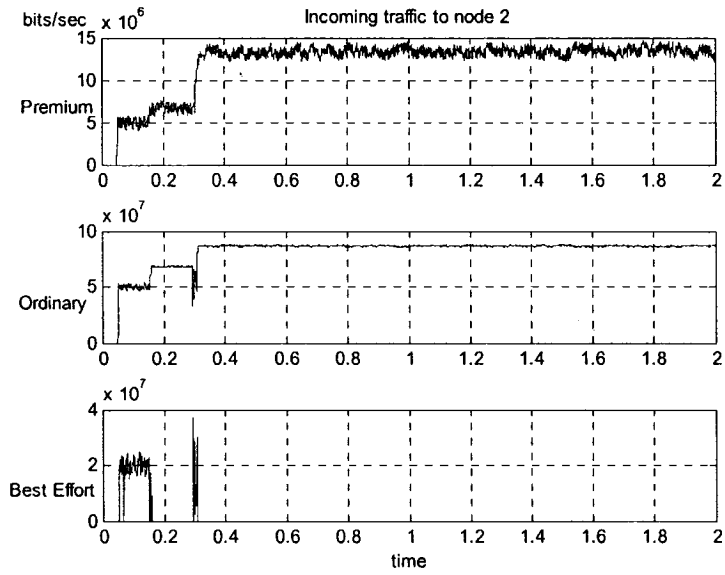

Figure 4-23: Actual incoming traffic to node 2

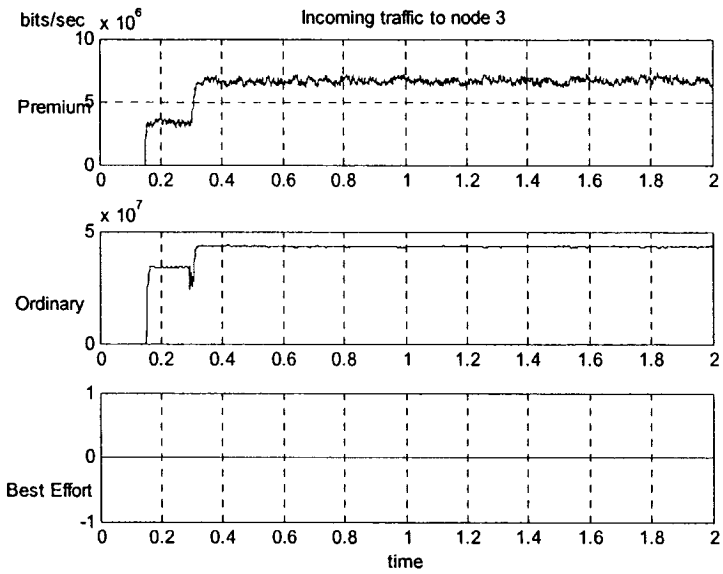

Figure 4-24: Actual incoming traffic to node 3

The queue length of each service in each node is shown in Figures 4-25, 4-26 and 4-27. We note that all queues have reached the preset values, so they are well controlled.

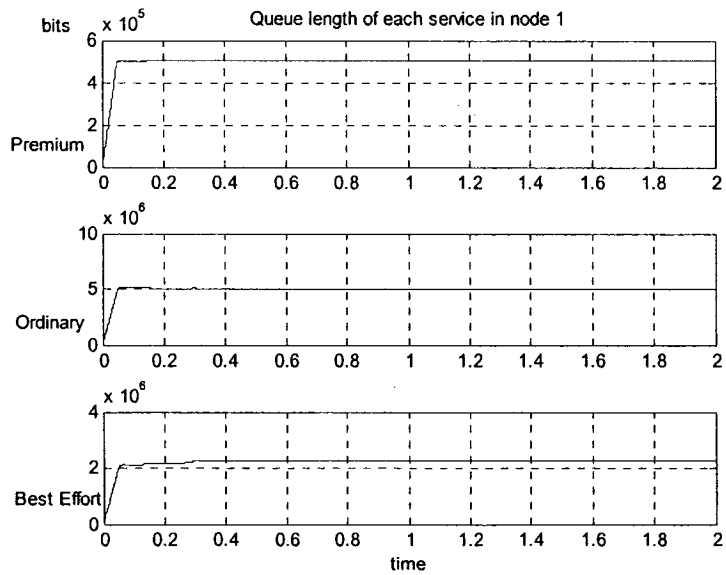

Figure 4-25: Queue length of each service in node 1

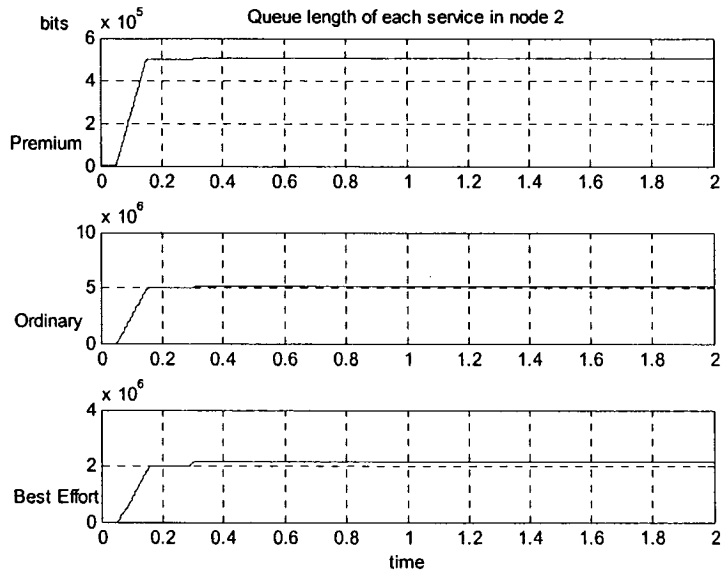

Figure 4-26: Queue length of each service in node 2

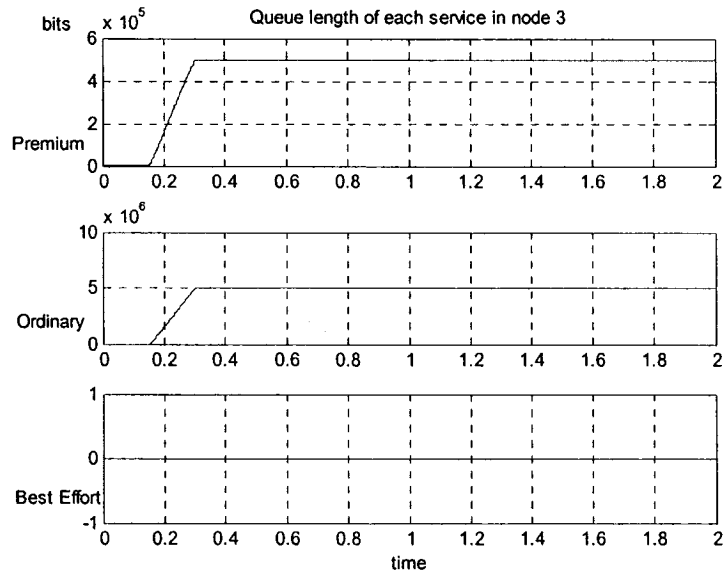

Figure 4-27: Queue length of each service in node 3

Figures 4-28, 4-29 and 4-30 show the bandwidth allocated to each service in each node. Figure 4-31 shows the total used bandwidth of each node. We notice that for nodes 1 and

2, the used bandwidth is the link capacity, which means the control algorithm works well. For node 3, there is no sufficient traffic, especially no feedback from other nodes, so the used bandwidth is far less than the link capacity.

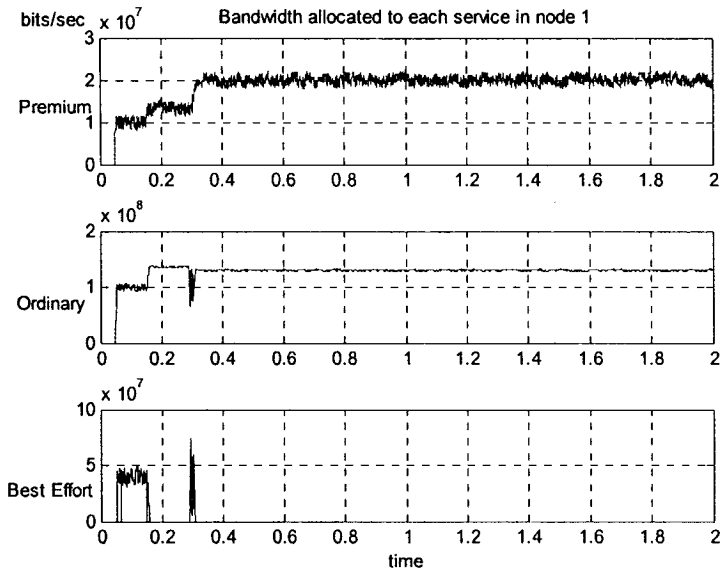

Figure 4-28: Bandwidth allocated to each service in node 1

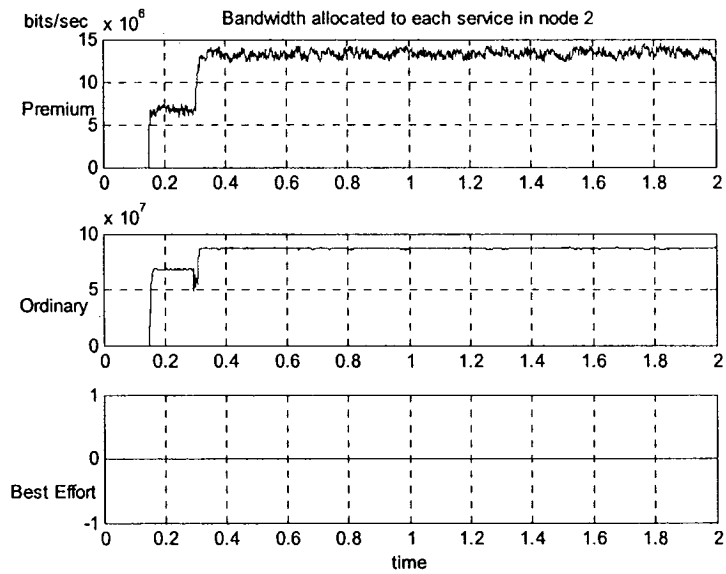

Figure 4-29: Bandwidth allocated to each service in node 2

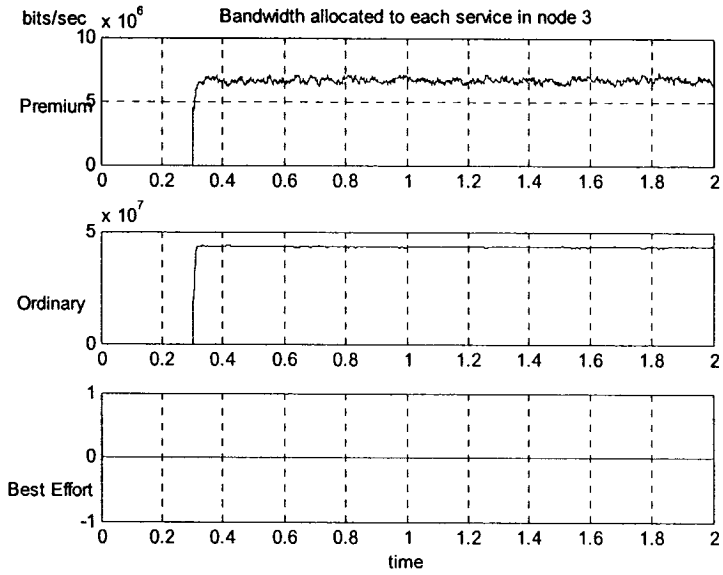

Figure 4-30: Bandwidth allocated to each service in node 3

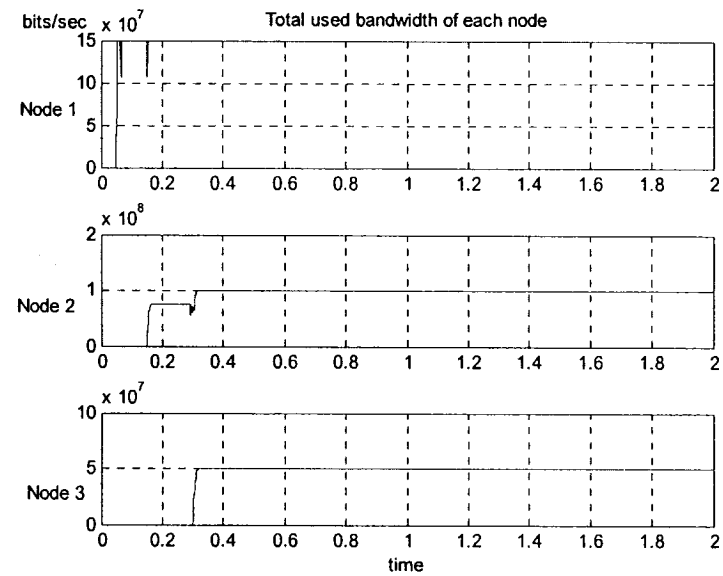

Figure 4-31: Total used bandwidth of each node

6. Simulation strategy 3

We keep all the parameters unchanged from incoming traffic 1, we generate the source traffic of each service continually as shown in Figure 4-3. And in the feedback channel, we select g_3 as shown in Figure 4-32.

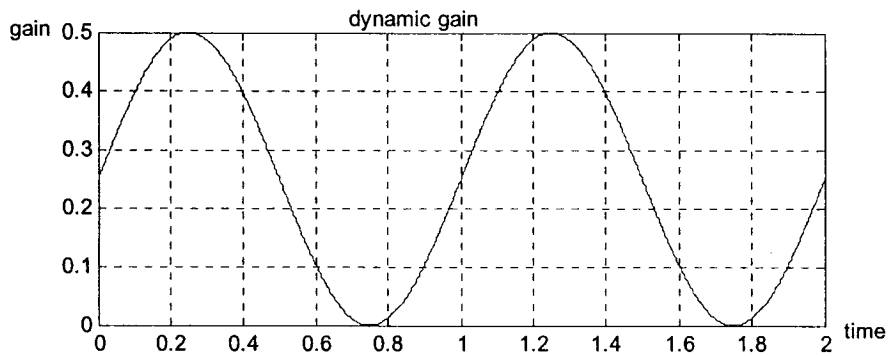

Figure 4-32: Dynamic feedback gain

At each node, we calculate the maximum allowed rate of ordinary traffic and best effort traffic that is shown in Figures 4-33 and 4-35. Because the feedback traffic is changing heavily according to the feedback gain, so the maximum allowed traffic rate of ordinary service in node 1 is also varying heavily. For node 2 and node 3, the vibration is smaller than node 1, because node 2 has a fixed feedback gain and node 3 has none.

In Figures 4-34 and 4-36, we compare the source traffic of ordinary and best effort service and the actual controlled rate (which is the smallest one of Figures 4-33 and 4-35). For ordinary service, we note that when the feedback gain is around 0 (at time 0.75s), the allowed rate reaches maximum, and the bandwidth is sufficient, so the leftover traffic can decrease to zero soon.

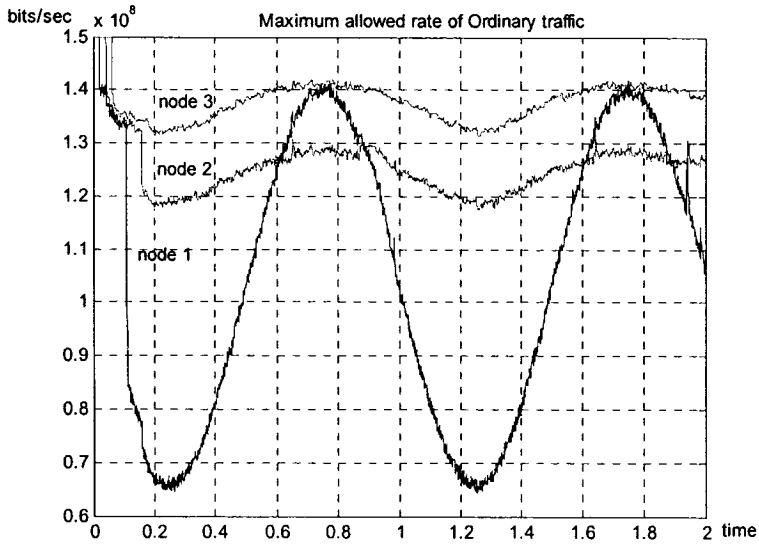

Figure 4-33: Maximum allowed rate of ordinary traffic

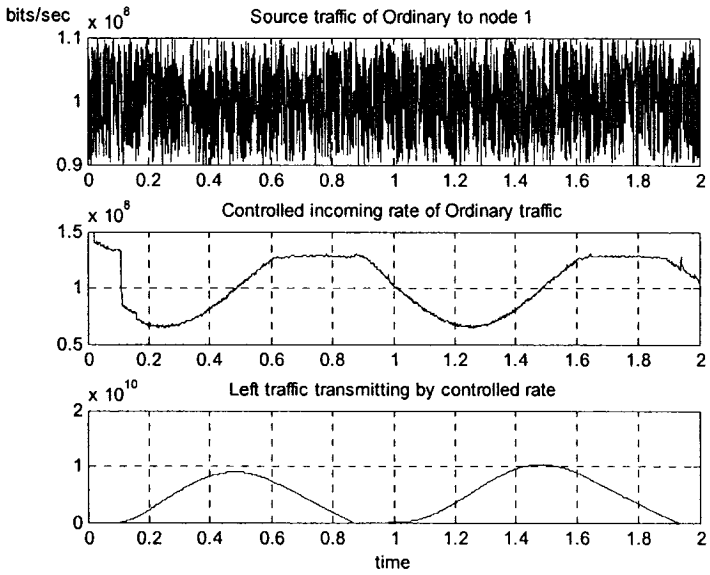

Figure 4-34: Source and actual rate and leftover traffic of ordinary service

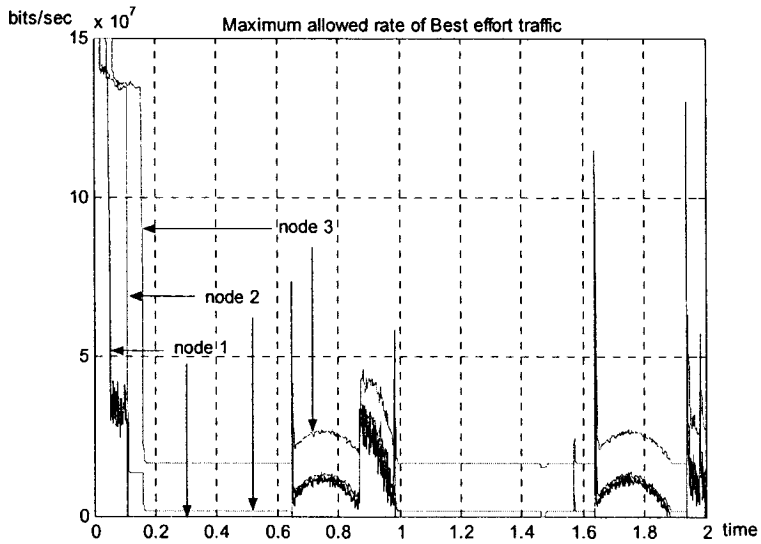

Figure 4-35: Maximum allowed rate of best effort traffic

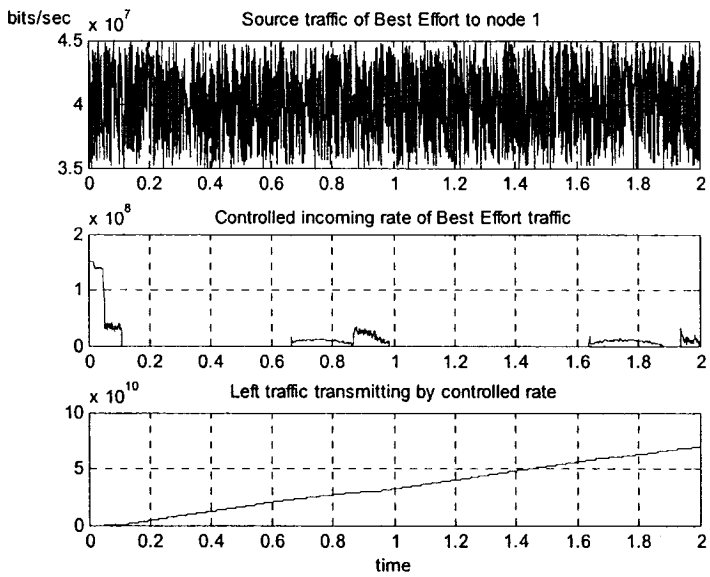

Figure 4-36: Source and actual rate and leftover traffic of best effort service

Figures 4-37, 4-38 and 4-39 show the jointed incoming traffic to nodes 1, 2 and 3, respectively. For premium service, it composes of the source traffic and feedback traffic

from nodes 2 and 3; for ordinary service and best effort service, it combines the controlled traffic and the feedback one. We note that for ordinary traffic and best effort rate matches the controlled rate (shown in Figures 4-34, 4-36 and 4-37). For node 2 (Figure 4-38), it receives the outgoing traffic of node 1 and feedback traffic from node 3. However, node 3 (Figure 4-39) only receives the traffic from node 2 without any feedback one.

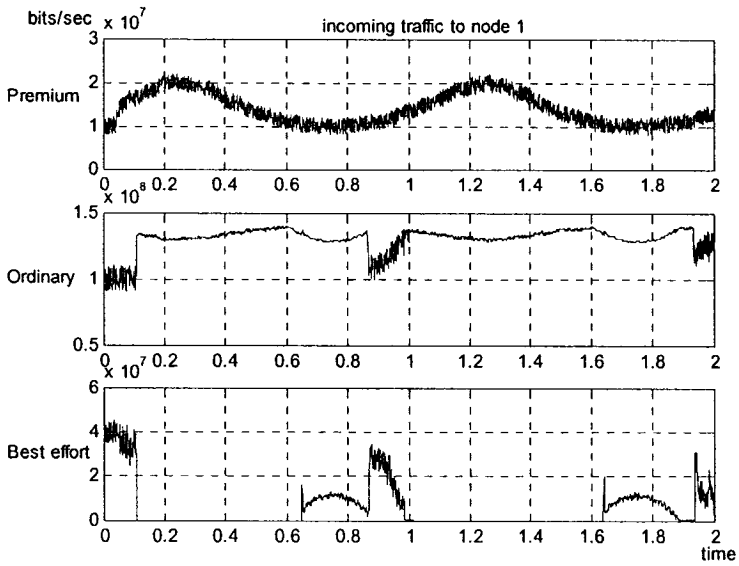

Figure 4-37: Actual incoming traffic to node 1

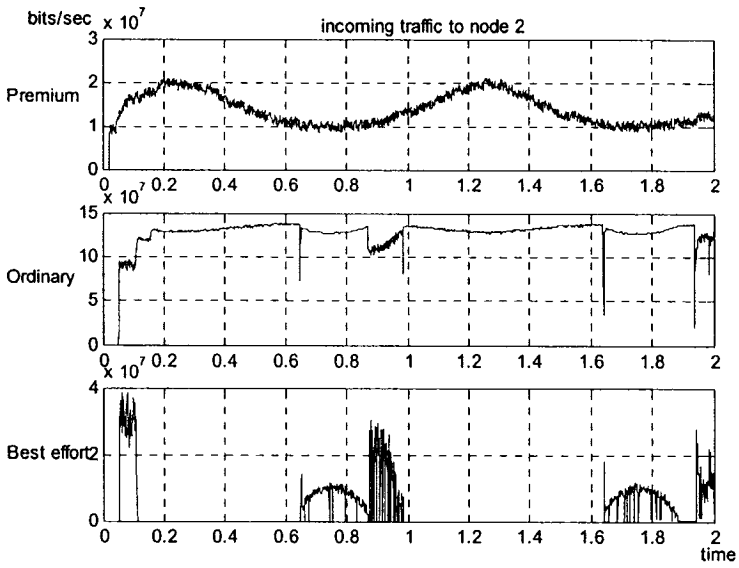

Figure 4-38: Actual incoming traffic to node 2

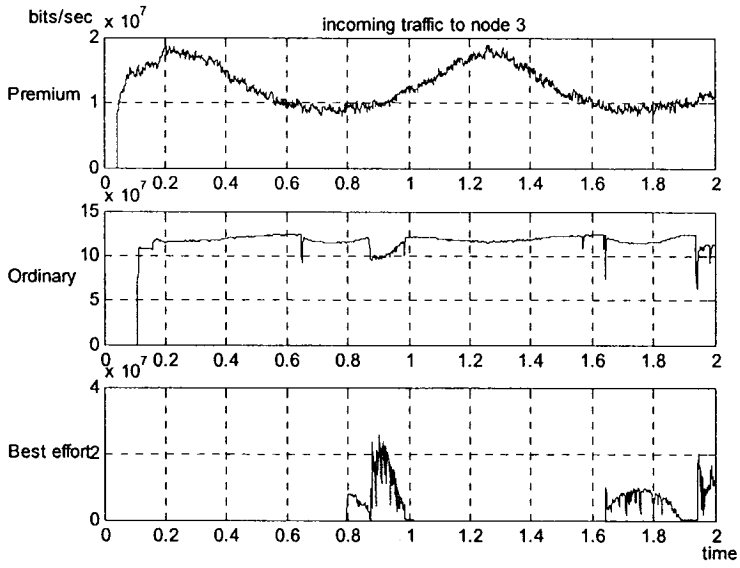

Figure 4-39: Actual incoming traffic to node 3

The queue length of each service in each node is shown in Figures 4-40, 4-41 and 4-42. We note that all queues have reached the preset value, so they are well controlled.

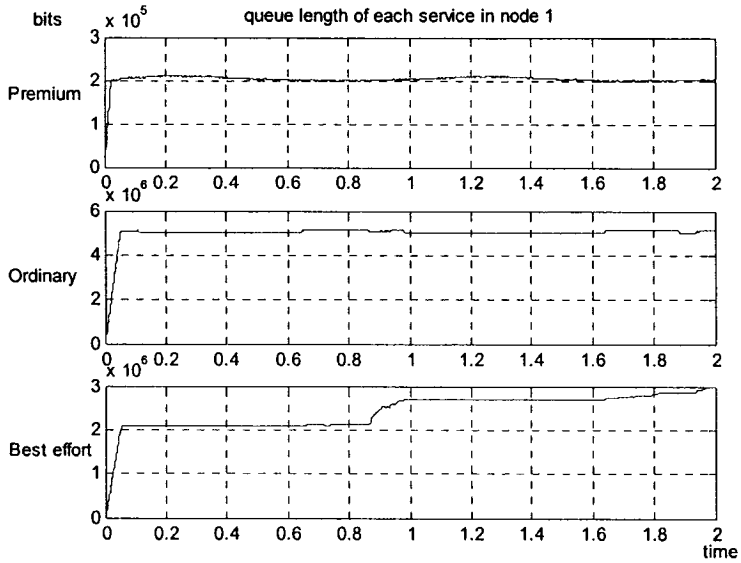

Figure 4-40: Queue length of each service in node 1

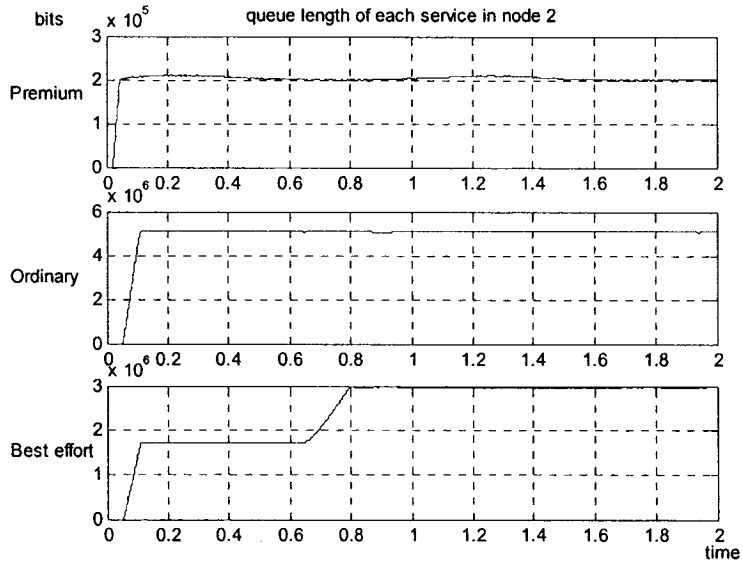

Figure 4-41: Queue length of each service in node 2

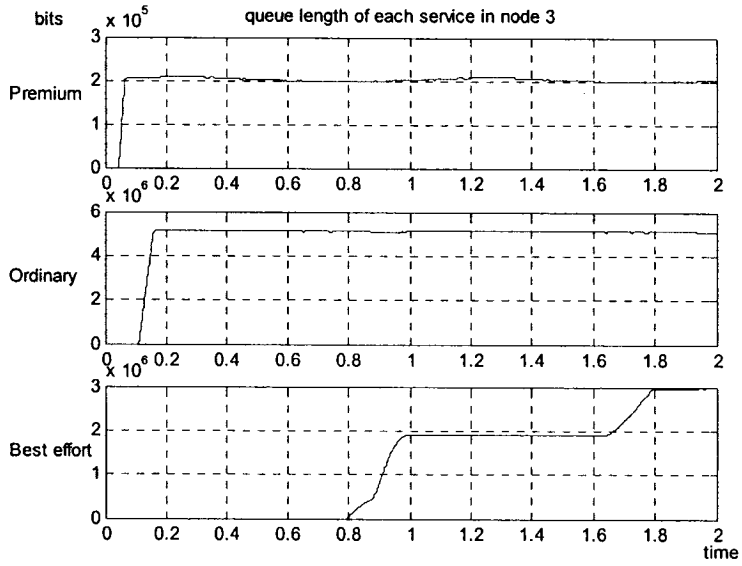

Figure 4-42: Queue length of each service in node 3

Figures 4-43, 4-44 and 4-45 show the bandwidth allocated to each service in each node. Figure 4-46 shows the total used bandwidth of each node. We notice that for nodes 1 and 2, the used bandwidth is the link capacity, which means the control algorithm works well. For node 3, there is no sufficient traffic, especially no feedback from other nodes, so the used bandwidth is far less than the link capacity.

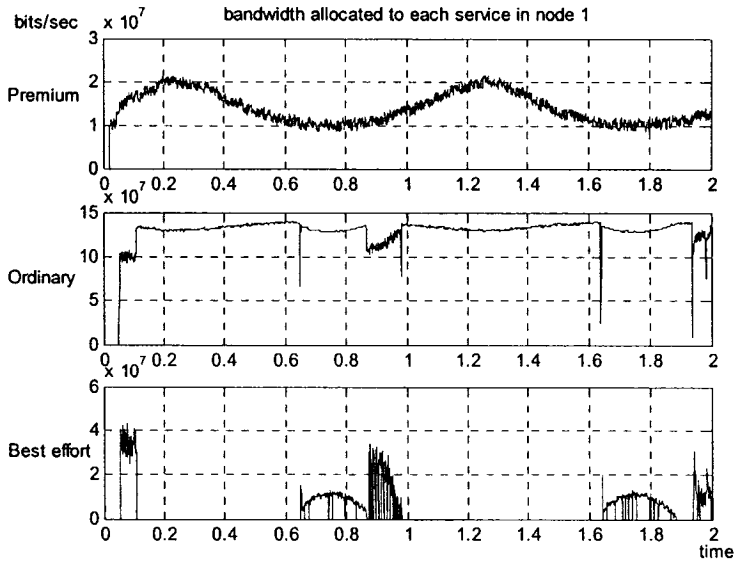

Figure 4-43: Bandwidth allocated to each service in node 1

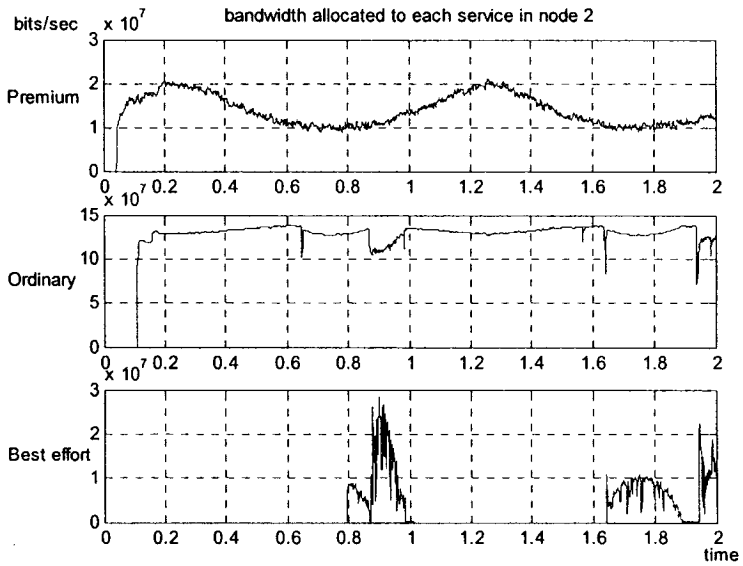

Figure 4-44: Bandwidth allocated to each service in node 2

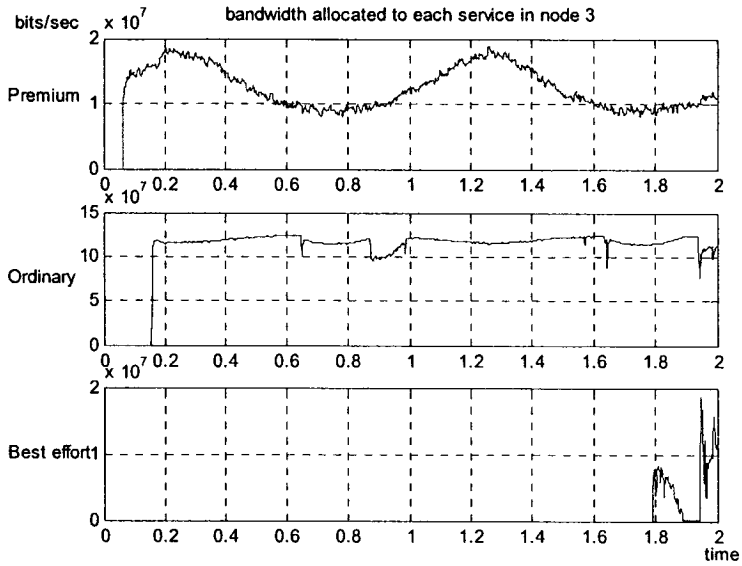

Figure 4-45: Bandwidth allocated to each service in node 3

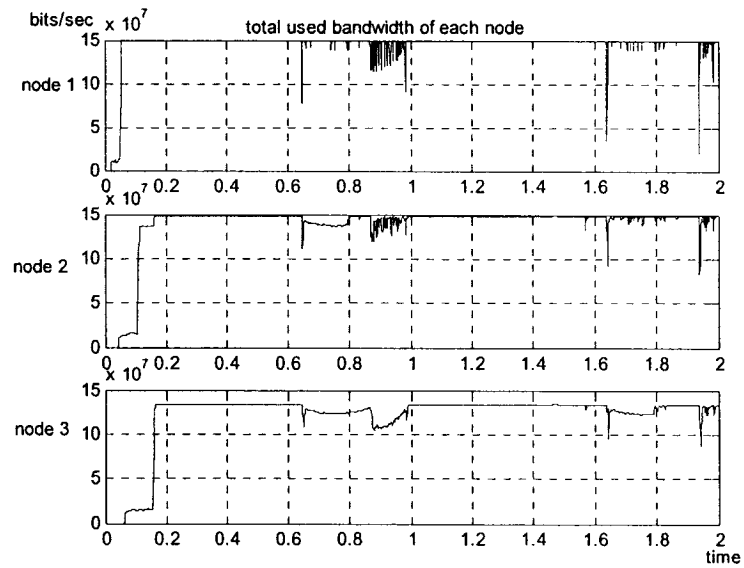

Figure 4-46: Total used bandwidth of each node

From the above simulation results, we can draw a conclusion that we achieve the desired

goal. For different simulation strategies (1, 2 3), the premium traffic is guaranteed to get enough bandwidth to pass though. For ordinary traffic, we investigate the model's performance by different parameters. We can see the traffic is under good control in both regular (simulation strategy 1) and congestion conditions (simulation strategy 2). For best effort traffic, the rate is also well controlled though the traffic is always in congestion in each strategy. We also investigate the model's dynamic capability by simulation strategy 3. It shows that our control objective is also achieved under a dynamic environment.

4.3 Centralized method

4.3.1 Designed algorithm and model

First, we do the research based on premium traffic among the three nodes. Based on Figures 4-1 and 4-2, we assume that all the delay units have the same value, τ; and the gain from node 1 to node 2 equals g_1, that from node 2 to node 3 equals g_2, and those of the feedback channels are g_3. Also $\lambda_{p1in}(t)=k_1, \lambda_{p2in}(t)=k_2, \lambda_{p3in}(t)=k_3$ are considered as external inputs.

Let $f_1(t)=\frac{x_1(t)}{1+x_1(t)}, f_2(t)=\frac{x_2(t)}{1+x_2(t)}, f_3(t)=\frac{x_3(t)}{1+x_3(t)}$, thus we obtain according to (4-1) to (4-12):

$$\begin{cases} \dot{x}_{p1} = \lambda_{p1in}(t) - f_1(t)C_1(t) + g_3 f_2(t-2\tau)C_2(t-2\tau) + g_3 f_3(t-3\tau)C_3(t-3\tau) \\ \dot{x}_{p2} = \lambda_{p2in}(t) + g_1 f_1(t-\tau)C_1(t-\tau) - f_2(t)C_2(t) + g_3 f_3(t-2\tau)C_3(t-2\tau) \\ \dot{x}_{p3} = \lambda_{p3in}(t) + g_2 f_2(t-\tau)C_2(t-\tau) - f_3(t)C_3(t) \end{cases} \quad (4\text{-}53)$$

We want to design a controller to achieve our goal, that is

$$\begin{cases} \dot{x}_{p1} = \alpha_{p1}\bar{x}_{p1}(t) = \alpha_{p1}(x_{p1}(t) - x_{p1}ref) \\ \dot{x}_{p2} = \alpha_{p2}\bar{x}_{p2}(t) = \alpha_{p2}(x_{p2}(t) - x_{p2}ref) \\ \dot{x}_{p3} = \alpha_{p3}\bar{x}_{p3}(t) = \alpha_{p3}(x_{p3}(t) - x_{p3}ref) \end{cases} \tag{4-54}$$

For the part in (4-44) including delay, we perform Taylor's approximation:

$$u(t-\tau) \xrightarrow{Laplace} u(s)e^{-s\tau} = u(s)(1 - s\tau + \frac{(s\tau)^2}{2!} - ...) \approx u(s)(1 - s\tau) \xrightarrow{Laplace^{-1}} u(t) - \tau\dot{u}(t)$$

Thus we get from (4-53)

$$\begin{cases} \dot{x}_{p1} = \lambda_{p1in}(t) - f_1(t)C_1(t) + g_3 f_2(t-2\tau)[C_2(t) - 2\tau\dot{C}_2(t)] \\ \qquad + g_3 f_3(t-3\tau)[C_3(t) - 3\tau\dot{C}_3(t)] \\ \dot{x}_{p2} = \lambda_{p2in}(t) + g_1 f_1(t-\tau)[C_1(t) - \tau\dot{C}_1(t)] - f_2(t)C_2(t) \\ \qquad + g_3 f_3(t-2\tau)[C_3(t) - 2\tau\dot{C}_3(t)] \\ \dot{x}_{p3} = \lambda_{p3in}(t) + g_2 f_2(t-\tau)[C_2(t) - \tau\dot{C}_2(t)] - f_3(t)C_3(t) \end{cases} \tag{4-55}$$

Let $\alpha_{p1} = \alpha_{p2} = \alpha_{p3} = \alpha$, we rewrite (4-54) and (4-55) by the state function

$$A * C + B * \dot{C} = D$$

$$\begin{bmatrix} -f_1(t) & g_3 f_2(t-2\tau) & g_3 f_3(t-3\tau) \\ g_1 f_1(t-\tau) & -f_2(t) & g_3 f_3(t-2\tau) \\ 0 & g_2 f_2(t-\tau) & -f_3(t) \end{bmatrix} \begin{bmatrix} C_1(t) \\ C_2(t) \\ C_3(t) \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & -2g_3\tau f_2(t-2\tau) & -3g_3\tau f_3(t-3\tau) \\ -g_1\tau f_1(t-\tau) & 0 & -2g_3\tau f_3(t-2\tau) \\ 0 & -g_2\tau f_2(t-\tau) & 0 \end{bmatrix} \begin{bmatrix} \dot{C}_1(t) \\ \dot{C}_2(t) \\ \dot{C}_3(t) \end{bmatrix} = \begin{bmatrix} \alpha\bar{x}_1 + k_1 \\ \alpha\bar{x}_2 + k_2 \\ \alpha\bar{x}_3 + k_3 \end{bmatrix} \tag{4-56}$$

According to (4-47), we notice that τ usually is a small value. So we assume $B \approx 0$, thus we get the control strategy:

$$A * C \approx D \Rightarrow C = A^{-1}D \tag{4-57}$$

$$A^{-1} = \frac{\begin{bmatrix} f_2(t)f_3(t-2\tau) - g_2g_3f_2(t-\tau)f_3(t-2\tau) & g_3f_3(t)f_2(t-2\tau) + g_2g_3f_2(t-\tau)f_3(t-3\tau) & g_3{}^2f_3(t-2\tau)f_2(t-2\tau) + g_3f_2(t)f_3(t-3\tau) \\ g_1f_1(t-\tau)f_3(t) & f_1(t)f_3(t) & g_3f_1(t)f_3(t-2\tau) + g_1g_3f_1(t-\tau)f_3(t-3\tau) \\ g_1g_2f_1(t-\tau)f_2(t-\tau) & g_2f_1(t)f_2(t-\tau) & f_1(t)f_2(t) - g_1g_3f_1(t-\tau)f_2(t-2\tau) \end{bmatrix}}{\begin{matrix} g_1g_3f_2(t-2\tau)f_1(t-\tau)f_3(t) + g_2g_3f_1(t)f_3(t-2\tau)f_2(t-\tau) \\ + g_1g_2g_3f_1(t-\tau)f_3(t-3\tau)f_2(t-\tau) - f_1(t)f_2(t)f_3(t) \end{matrix}} \quad (4\text{-}58)$$

The control objective in each node is to choose the capacity $C_p(t)$ to be allocated to the traffic under the constraint that the incoming traffic rate $\lambda_p(t)$ is unknown but bounded, so that the averaged buffer size $x_p(t)$ is as close to the desired value x^{ref} (chosen by the operator or designer) as possible. The algorithm is described in Section 2.1.

Assuming the ordinary and best effort traffic are sufficient to cause congestion, so we can just regulates the flow rate of ordinary traffic and best effort traffic into the network, by monitoring the length of the queue and the available capacity. At each updated time unit, we find the maximum allowed packets rate $\lambda_{o\max}(t)$ and $\lambda_{b\max}(t)$ (calculated based on (4-13) to (4-38)) for each node, and then we choose the smallest one as the source-outgoing rate of each service.

4.3.2 Stability analysis

When the system reaches its equilibrium, $\dot{\bar{x}}$ in (4-53) will approximate 0, which means $x \to x^{ref}$. In such a condition, we can assume $f(t) \approx 1$ for easy calculation.

Substituting (4-58) into (4-53) we will get:

$$\begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{p3} \end{bmatrix} = \begin{bmatrix} \alpha_{p1} & & 0 \\ & \alpha_{p2} & \\ 0 & & \alpha_{p3} \end{bmatrix} \begin{bmatrix} x_{p1}(t) - x_{p1}ref \\ x_{p2}(t) - x_{p2}ref \\ x_{p3}(t) - x_{p3}ref \end{bmatrix} \qquad (4\text{-}59)$$

It's easy to show that $\dot{\bar{x}}$ is bounded when we choose $\alpha_{p1}, \alpha_{p2}, \alpha_{p3} < 0$.

4.3.3 Simulation method and result

Based on this model, we perform the following simulations and obtain the results shown below.

7. Simulation strategy 1

We choose $g_1 = g_2 = g_3 = 0.5$, $\alpha_p = -500$, $\tau = 1ms$, $x_p buffer = 5\text{M}bits$, $x_p ref = 500\text{K}bits$, $k_1 = 10 \times 10^6, k_2 = k_3 = 0$ and other parameters are kept unchanged. Then we generate the same incoming traffic as shown in Figure 4-3.

At each node, we calculate the maximum allowed rate of ordinary traffic and best effort traffic which is shown in Figures 4-47 and 4-49. In Figures 4-48 and 4-50, we compare the source traffic of ordinary and best effort service with the actual controlled rate (which

is the smallest one of Figures 4-47 and 4-49), we note that although the incoming traffic is heavy, but the maximum allowed rate is limited. There is insufficient bandwidth for both ordinary and best effort service, so congestion occurs (shown in Figures 4-48 and 4-50), which means incoming packets are hold before flowing into node 1.

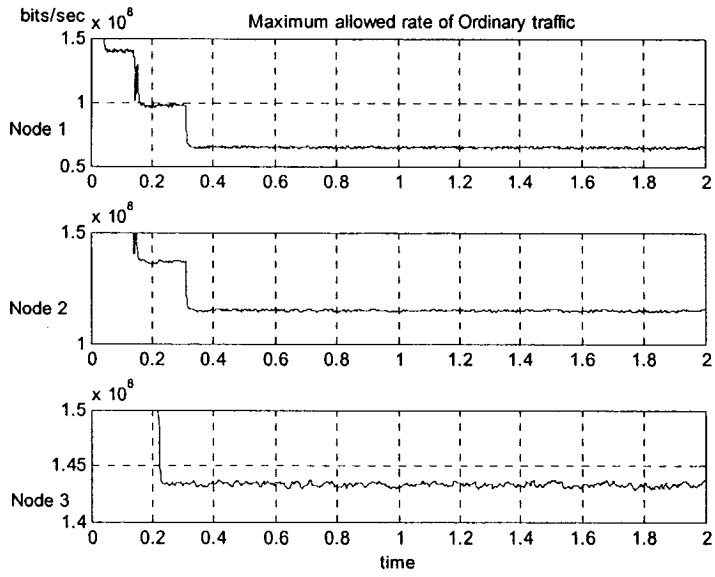

Figure 4-47: Maximum allowed rate of ordinary traffic

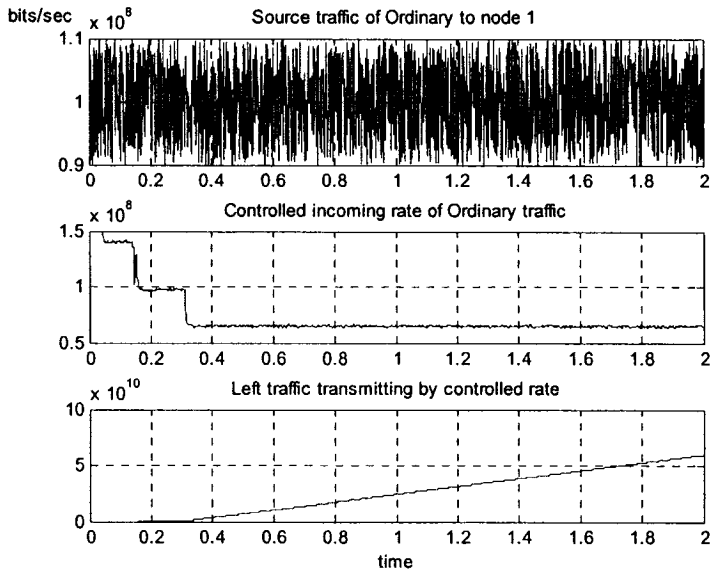

Figure 4-48: Source and actual rate and leftover traffic of ordinary service

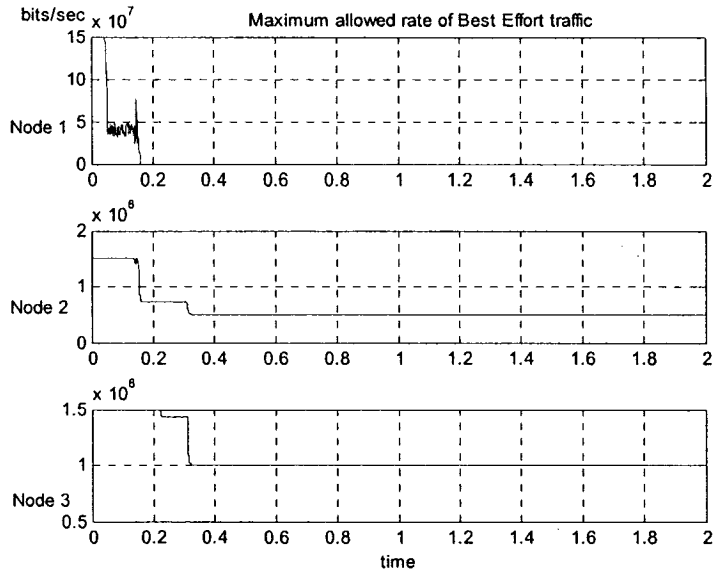

Figure 4-49: Maximum allowed rate of best effort traffic

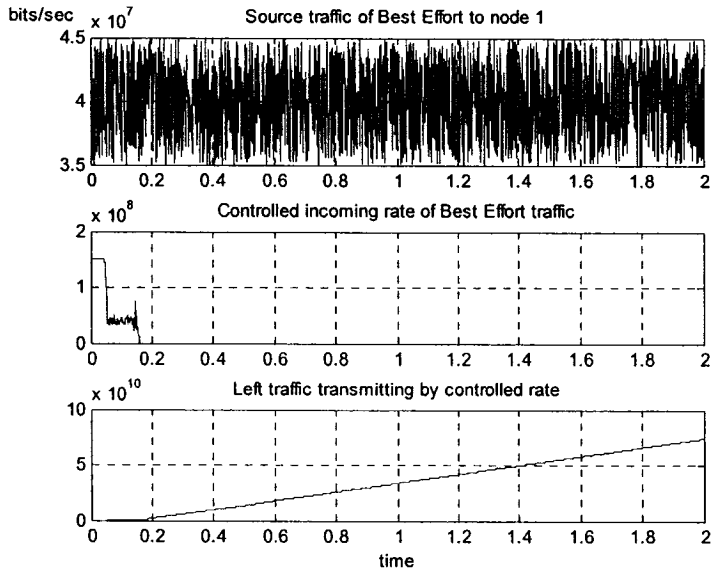

Figure 4-50: Source and actual rate and leftover traffic of best effort service

Figures 4-51, 4-52 and 4-53 show the jointed incoming traffic to nodes 1, 2 and 3, respectively. For premium service, it composes of the source traffic and feedback traffic from node 2; for ordinary service and best effort service, it combines the controlled traffic and the feedback one. We note that ordinary and best effort rate matches the controlled rate shown in Figures 4-47 and 4-49. For node 2 (Figure 4-52), it receives the outgoing traffic of node 1 and feedback traffic from node 3. However, node 3 (Figure 4-53) only accepts the traffic from node 2 without any feedback one.

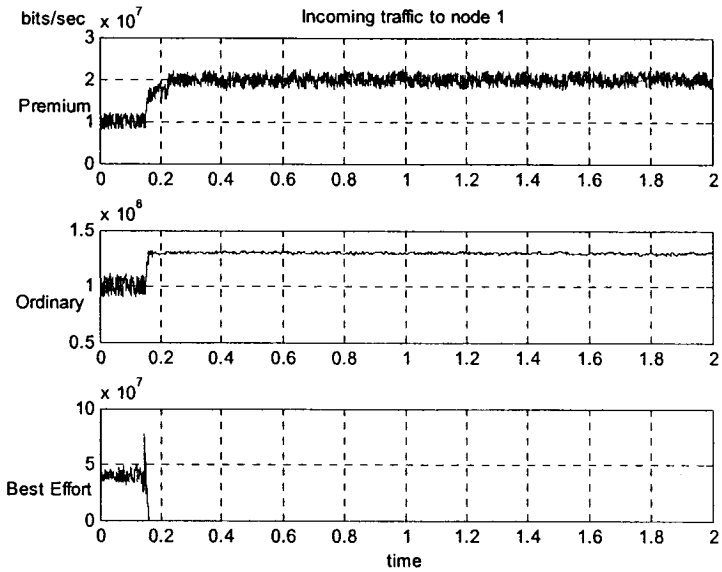

Figure 4-51: Actual incoming traffic to node 1

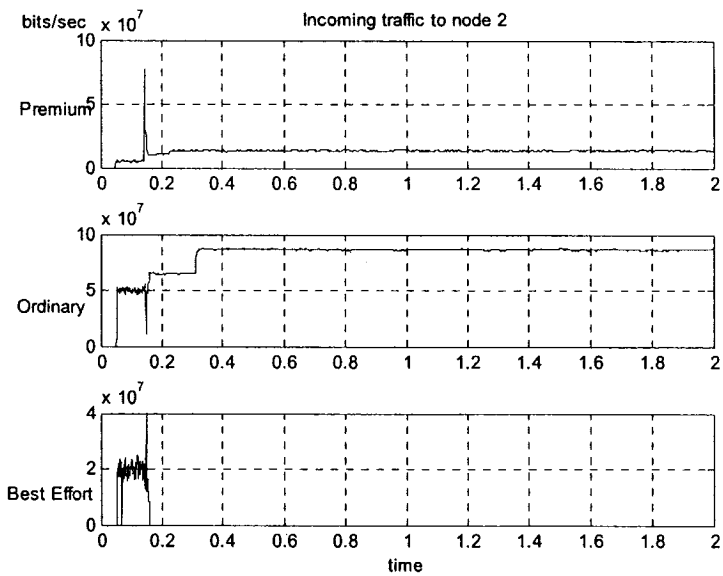

Figure 4-52: Actual incoming traffic to node 2

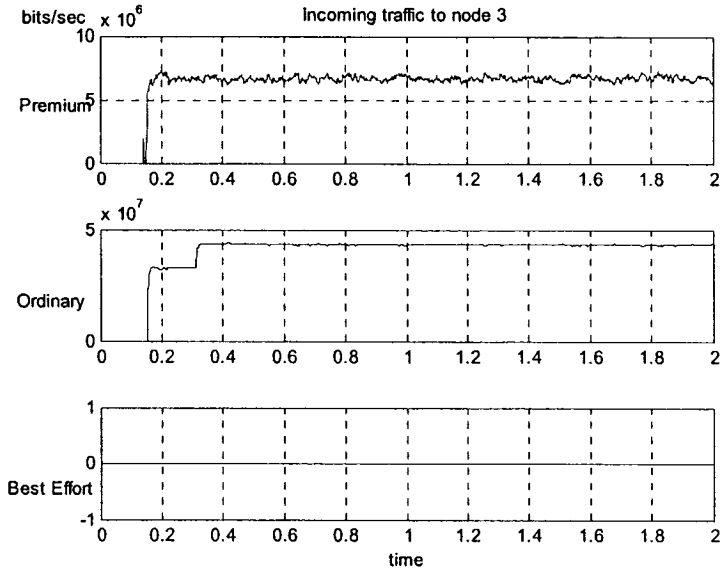

Figure 4-53: Actual incoming traffic to node 3

The queue length of each service in each node is shown in Figures 4-54, 4-55 and 4-56. We note that all queues have reached the preset value, so they are well controlled.

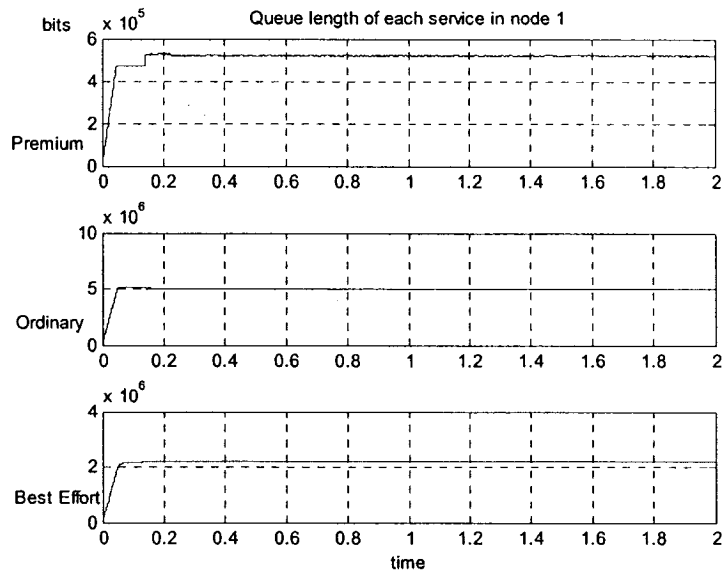

Figure 4-54: Queue length of each service in node 1

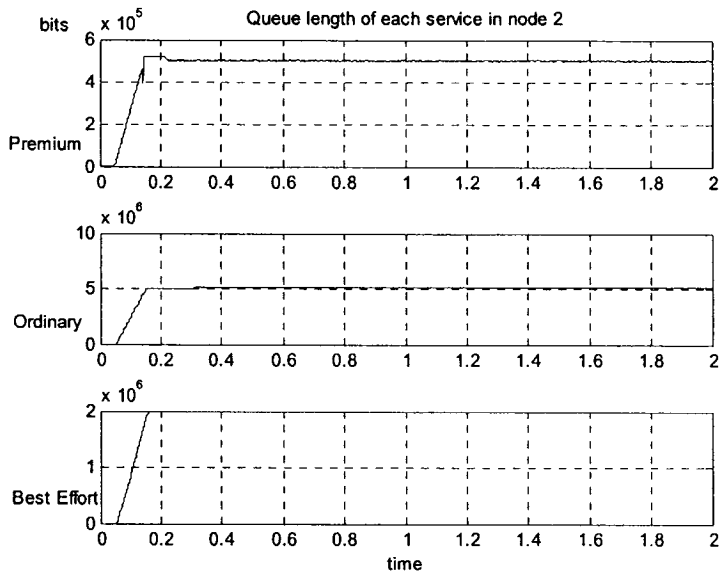

Figure 4-55: Queue length of each service in node 2

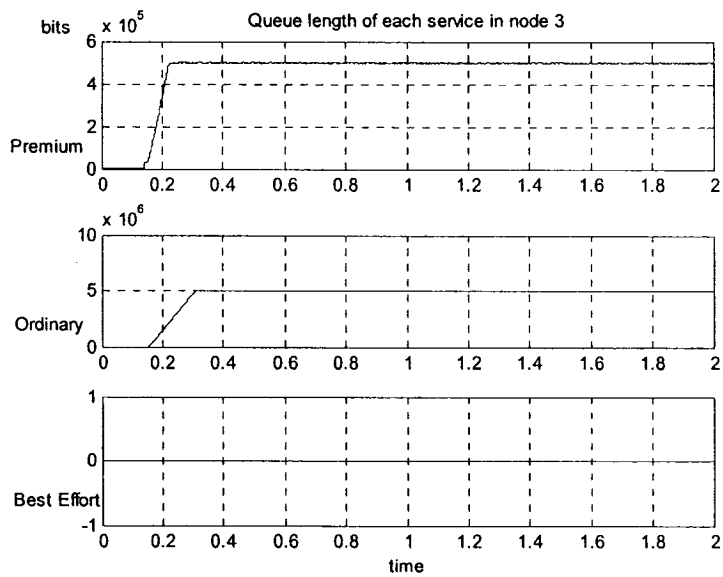

Figure 4-56: Queue length of each service in node 3

Figures 4-57, 4-58 and 4-59 show the bandwidth allocated to each service in each node.

Figure 4-60 shows the total used bandwidth of each node. We notice that for nodes 1 and 2, the used bandwidth is the link capacity, which means the control algorithm works well. For node 3, there is no sufficient traffic, especially no feedback from other nodes, so the used bandwidth is far less than the link capacity.

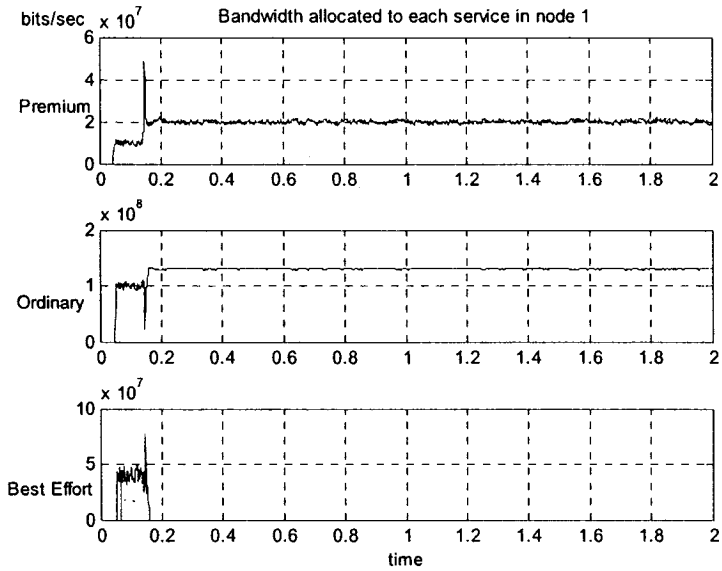

Figure 4-57: Bandwidth allocated to each service in node 1

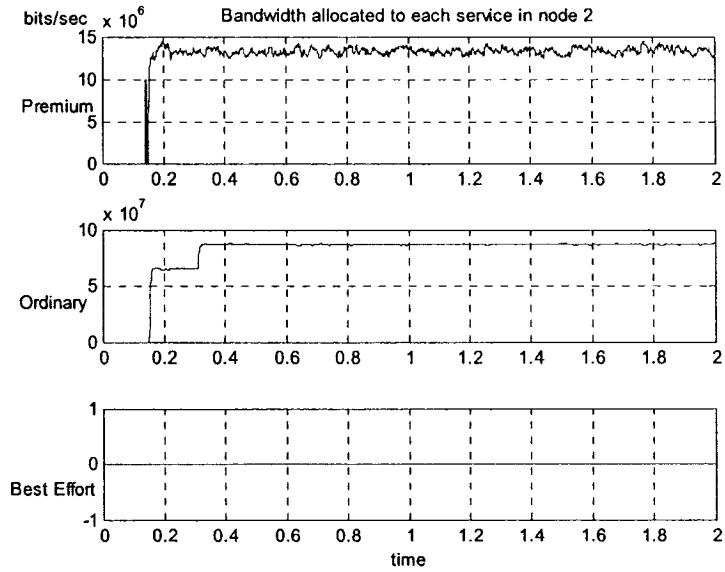

Figure 4-58: Bandwidth allocated to each service in node 2

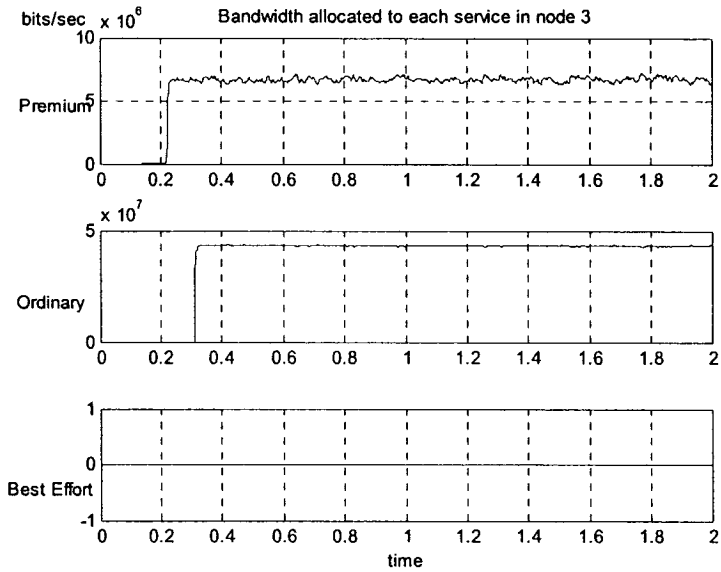

Figure 4-59: Bandwidth allocated to each service in node 3

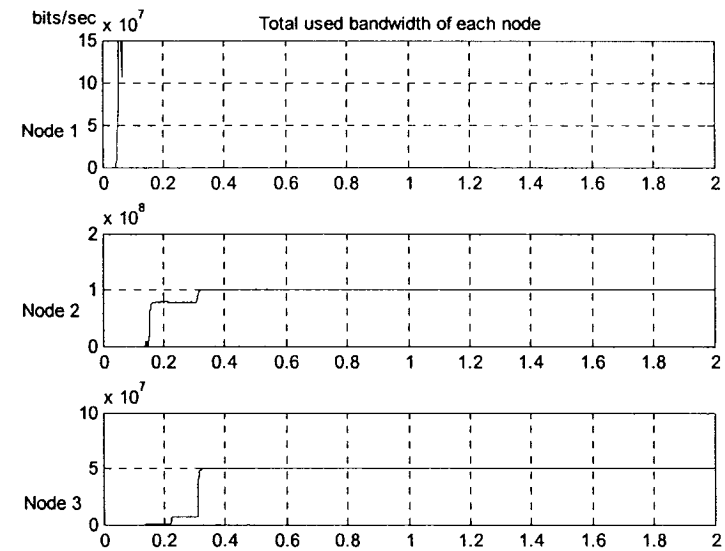

Figure 4-60: Total used bandwidth of each node

8. Simulation strategy 2

We keep all the parameters unchanged and we choose $g_1 = g_2 = 0.9$, $g_2 = 0.1$ and $x_p ref = 200\text{K}bits$, and in the feedback channel we select g_3 as shown in Figure 4-32. We generate the source traffic of each service continually as shown in Figure 4-3.

At each node, we calculate the maximum allowed rate of ordinary traffic and best Effort traffic that is shown in Figures 4-61 and 4-63. Because the feedback traffic is changing heavily according to the feedback gain, the maximum allowed traffic rate of ordinary service in node 1 is also varying heavily. For node 2 and node 3, the oscillation is smaller than node 1, because node 2 has a fixed feedback gain and node 3 has none.

In Figures 4-62 and 4-64, we compare the source traffic of ordinary and best effort service and the actual controlled rate (which is the smallest one of Figures 4-61 and 4-63). For ordinary service, we note that when the feedback gain is around 0 (at time 0.75s), the allowed rate reaches maximum, and the bandwidth is sufficient, so the leftover traffic can decrease to zero quickly.

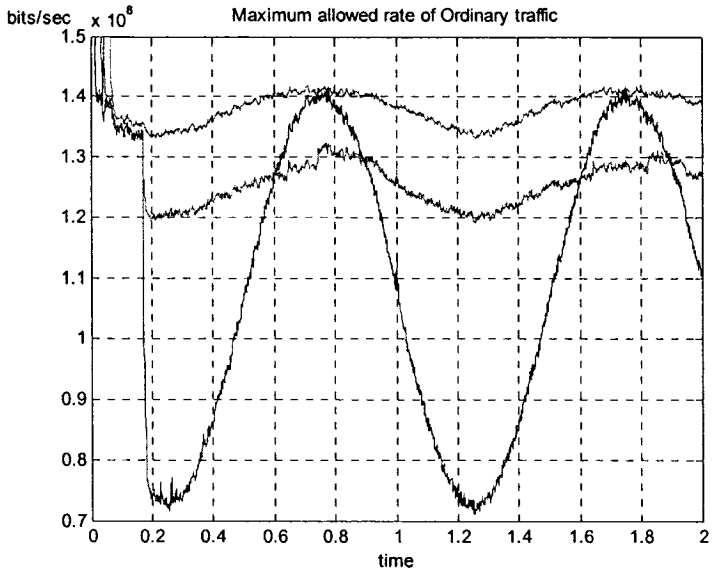

Figure 4-61: Maximum allowed rate of ordinary traffic

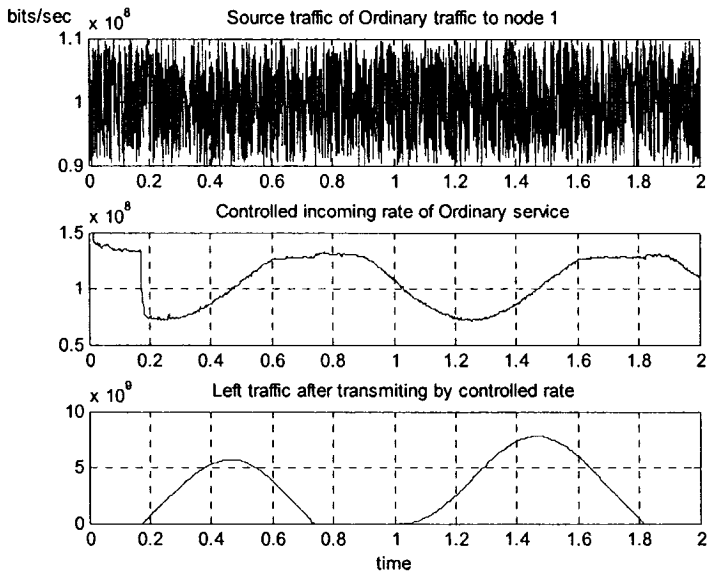

Figure 4-62: Source and actual rate and leftover traffic of ordinary service

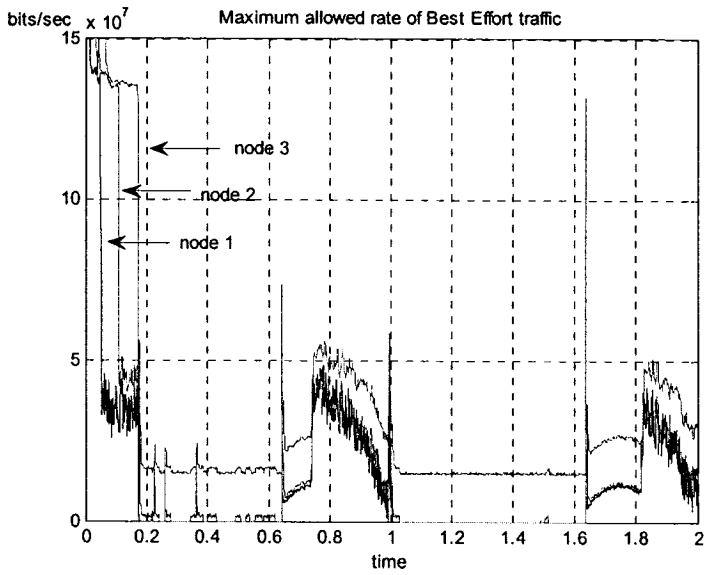

Figure 4-63: Maximum allowed rate of best effort traffic

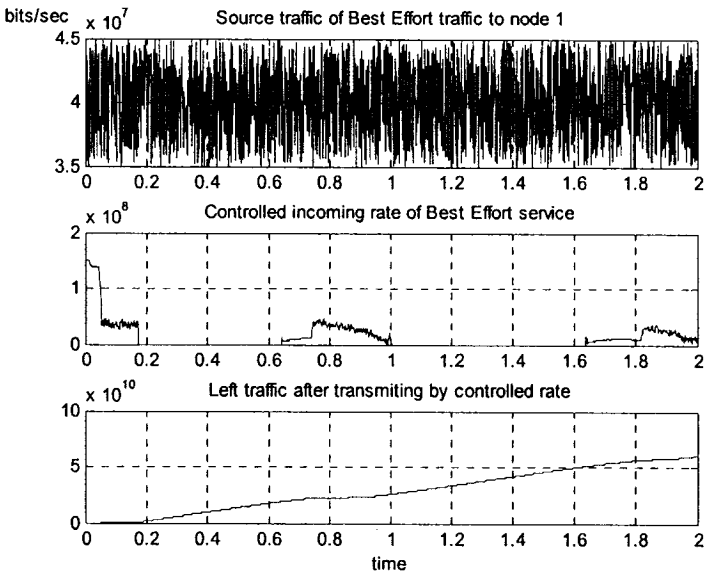

Figure 4-64: Source and actual rate and leftover traffic of best effort service

Figures 4-65, 4-66 and 4-67 show the jointed incoming traffic to nodes 1, 2 and 3, respectively. For premium service, it is composed of the source traffic and feedback traffic from nodes 2 and 3; for ordinary service and best effort service, it combines the controlled traffic and the feedback one. We note that for ordinary traffic and best effort rate matches the controlled rate (shown in Figures 4-62, 4-64 and 4-65). For node 2 (Figure 4-66), it receives the outgoing traffic of node 1 and feedback traffic from node 3. However, node 3 (Figure 4-67) only receives the traffic from node 2 without any feedback one.

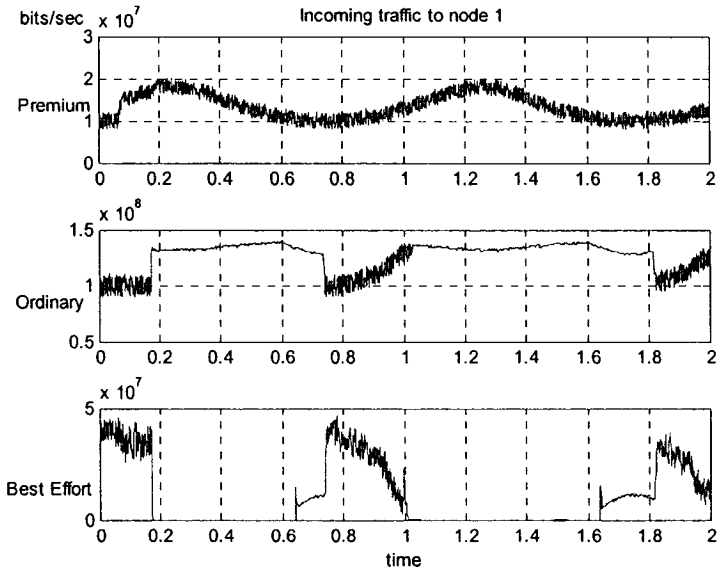

Figure 4-65: Actual incoming traffic to node 1

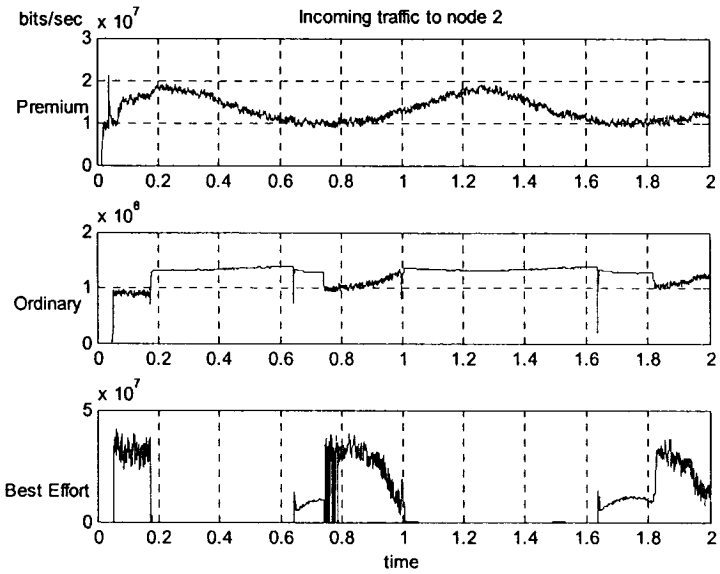

Figure 4-66: Actual incoming traffic to node 2

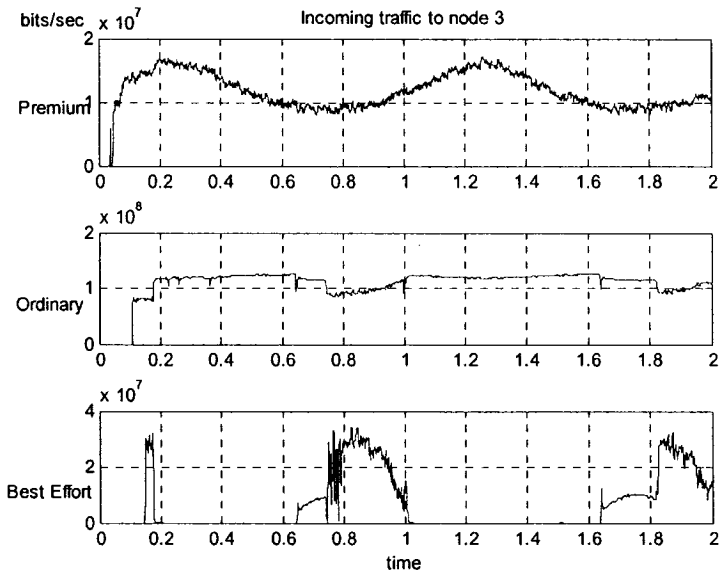

Figure 4-67: Actual incoming traffic to node 3

The queue length of each service in each node is shown in Figures 4-68, 4-69 and 4-70. We note that all queues have reached the preset value, so they are well controlled.

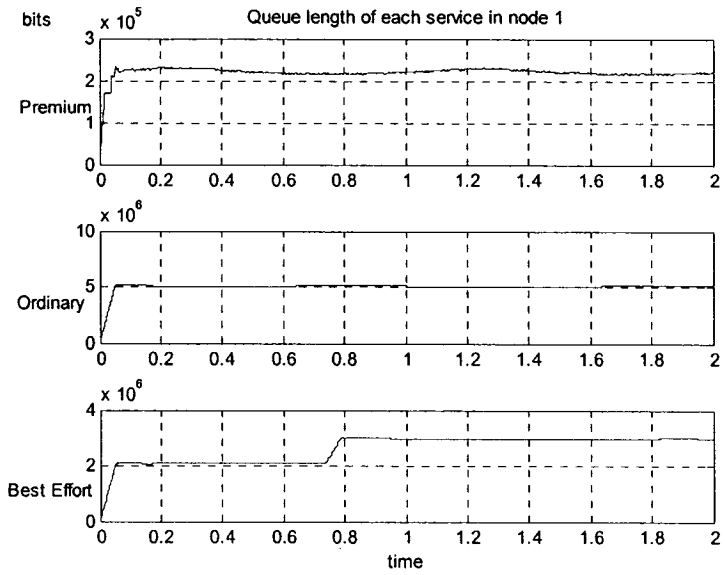

Figure 4-68: Queue length of each service in node 1

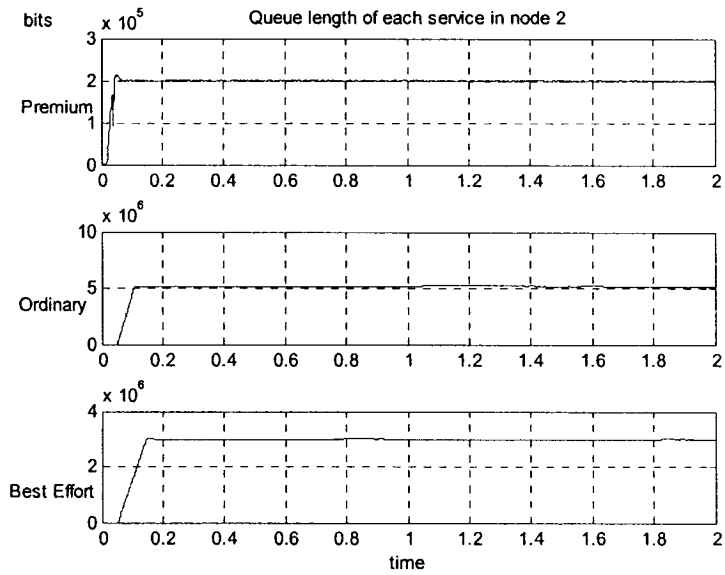

Figure 4-69: Queue length of each service in node 2

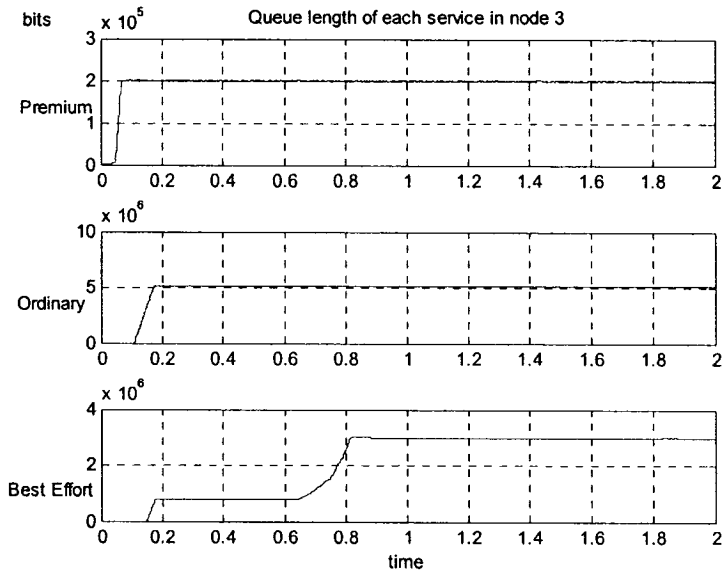

Figure 4-70: Queue length of each service in node 3

Figures 4-71, 4-72 and 4-73 show the bandwidth allocated to each service in each node. Figure 4-74 shows the total used bandwidth of each node. We notice that for nodes 1 and 2, the used bandwidth is the link capacity, which means the control algorithm works well. For node 3, there is no sufficient traffic, especially no feedback from other nodes, so the used bandwidth is far less than the link capacity.

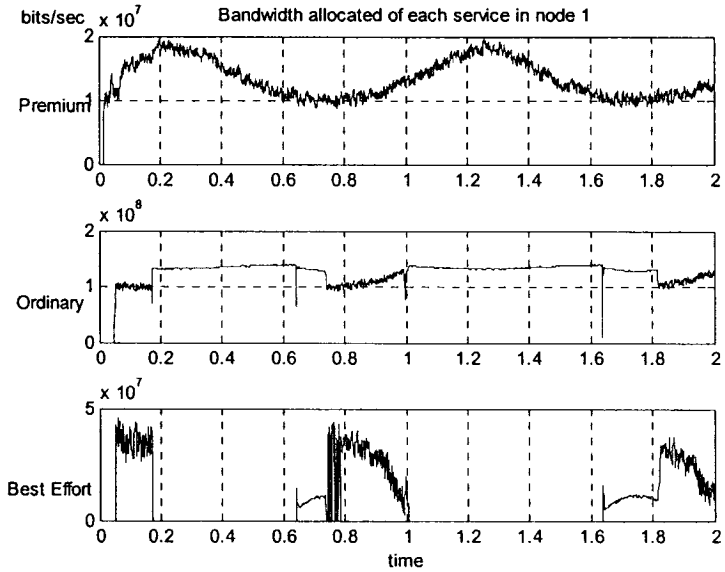

Figure 4-71: Bandwidth allocated to each service in node 1

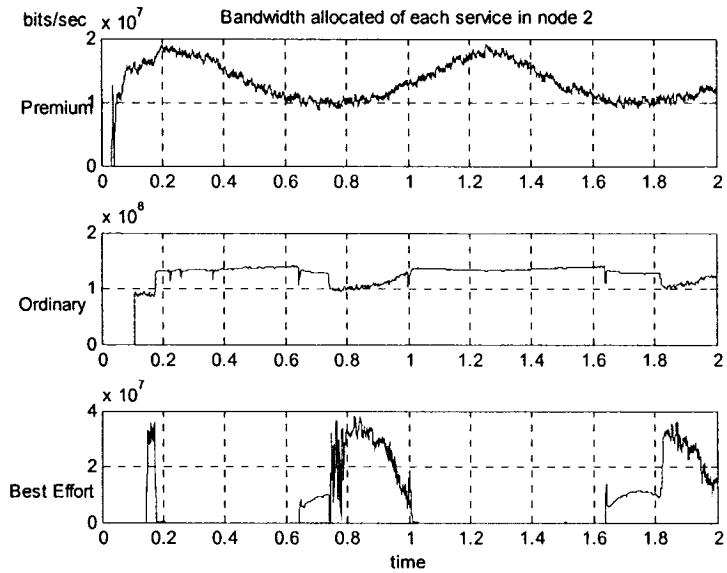

Figure 4-72: Bandwidth allocated to each service in node 2

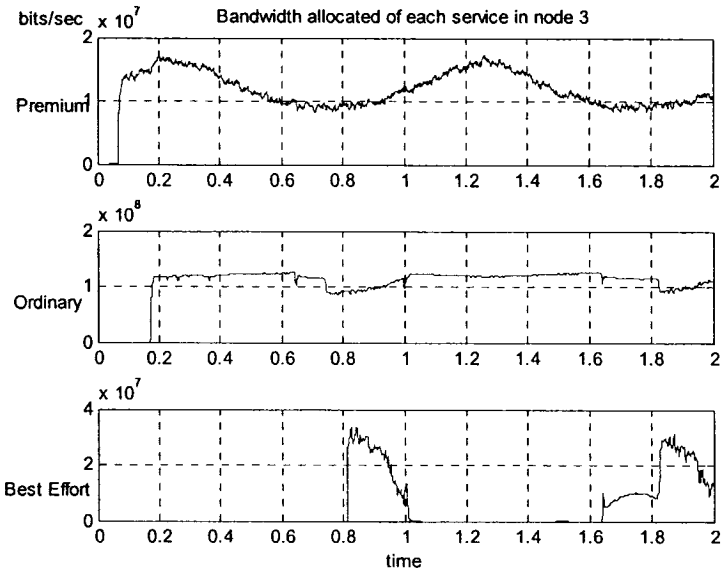

Figure 4-73: Bandwidth allocated to each service in node 3

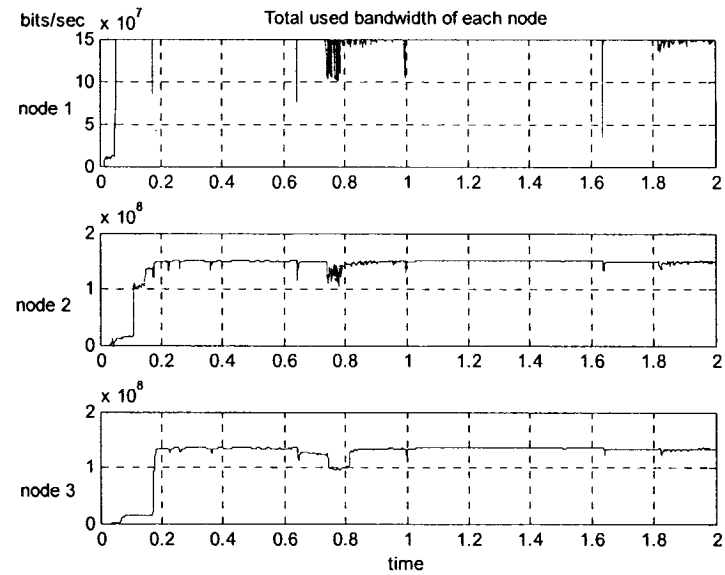

Figure 4-74: Total used bandwidth of each node

From the above simulation results, we can draw a conclusion that we achieved the desired goal. For different simulation strategy (1 and 2), the premium traffic is guaranteed

to get enough bandwidth to pass though. For ordinary and best effort traffic, we can see the traffic is under good control in congestion condition (simulation strategy 1 and 2). We also investigate the model's dynamic capability by simulation strategy, it shows that our control objective is also achieved under a dynamic environment.

4.4 Centralized method Vs Decentralized method

In this section, we compare the decentralized method with centralized one from the following perspectives: effect of delay and performance according to different gains.

4.4.1 Effect of delay

In this section, we investigate how the delay affects the system's performance. First, for the decentralized method we change the value of the gains, to see how much delay the system can tolerate. First, we choose $\alpha = 500$ and keep the other parameters unchanged from previous section and define the delay as an integer. Let $g_1 = g_2 = g_3$, we do the following three investigations.

Case I: we select low gains, e.g. $g_1 = g_2 = g_3 = 0.1$, then we calculate τ according to criteria (4-49) to (4-52) to get the $\tau_{\max} = 8ms$

Case II: we select medium gains, e.g. $g_1 = g_2 = g_3 = 0.5$, then we get the $\tau_{\max} = 1ms$

Case III: we select high gains, e.g. $g_1 = g_2 = g_3 = 0.9$, then we find $\tau_{\max}$ does not exist in integer format.

Case IV: we select different gains, e.g. $g_1 = 0.1; g_2 = 0.5; g_3 = 0.9$, then we calculate τ according to criteria (4-40) to (4-43) to get the $\tau_{max} = 1ms$

So based on cases I, II, III, we know the relation between gain and delay can be described as shown in the following figure. This means the tolerated delay decreases while the gain increases.

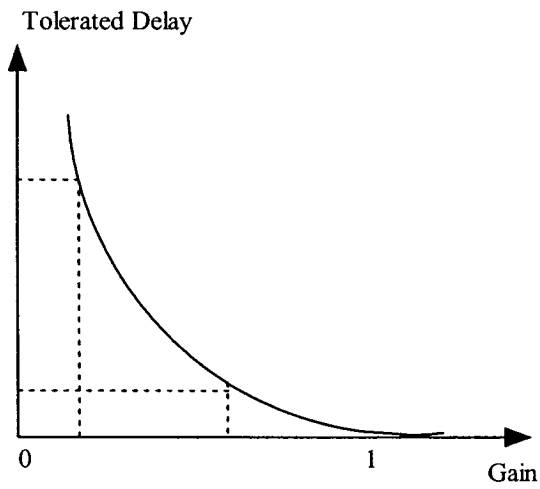

Figure 4-75: Relation between gain and delay

We now select $g_1 = g_2 = g_3 = 0.5$, $x_p ref = 500K bits$, and choose a large amount of delay, e.g., 1s, to simulate the model and the results are shown in Figure 4-76. Note that only the queue length of the premium is studied here. We can conclude that the results are not acceptable.

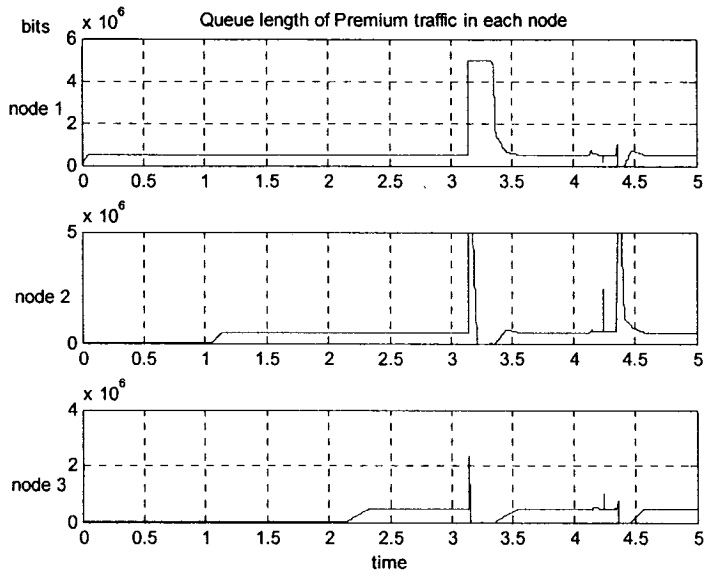

Figure 4-76: Queue length of premium traffic

Second, for the centralized method, we choose the same amount of delay, e.g., 1*s*, to simulate the system. Figure 4-77 shows the queue length of the premium traffic in each node. We find the system has a good performance. Hence the large amount of delay (1ms to 1000ms) nearly doesn't affect the system's performance.

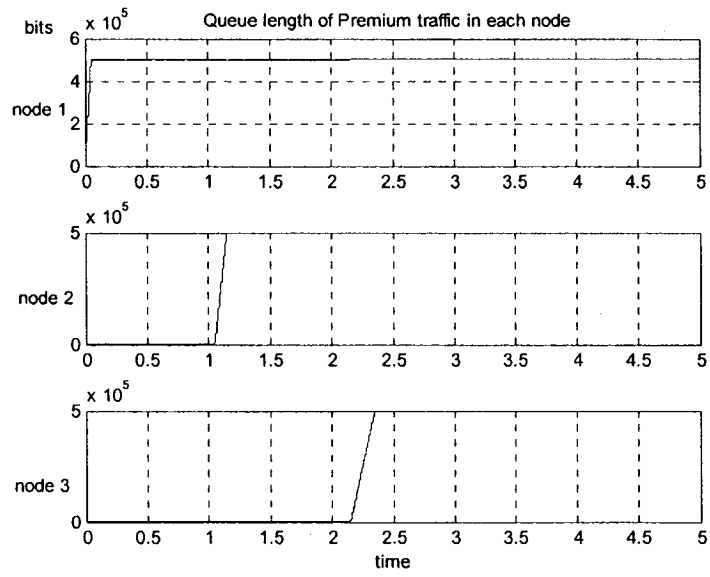

Figure 4-77: Queue length of premium traffic

Comparing the above results, we can draw a conclusion that the centralized model can tolerate more delay than the decentralized one.

4.4.2 Performance by different gains

In this section, we investigate how the gains affect the system's performance. We select different group of gains to simulate the two models and get the results as shown in the following three tables. We keep the other parameters unchanged from the previous section and define the delay as $1ms$.

Table 1: Performance results using different gains

$g_1 = g_2 = 0.9,\ g_3 = 0.1$ Decentralized									
Node	Overshoot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	0	0	0	0.05	0.052	0.4	0.2%	3%	-0.4%
2	0	0	0	0.107	0.1	0.47	0.2%	3%	-0.5%
3	0	0	0	0.165	0.164	0.5	0.3%	3.3%	-0.5%
$g_1 = g_2 = 0.9,\ g_3 = 0.1$ Centralized									
Node	Overshoot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	8%	0	0	0.12	0.052	0.39	4%	0.2%	0
2	4%	0	0	0.125	0.107	0.395	0.1%	2.5%	-0.6%
3	0	0	0	0.128	0.18	0.472	0.1%	2%	-0.55%

Table 2: Performance results using different gains (con. 1)

$g_1 = g_2 = g_3 = 0.5$ Decentralized									
Node	Overshot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	0	0	N/A	0.05	0.052	N/A	1%	0	N/A
2	0	0	N/A	0.15	0.152	N/A	0.3%	1.6%	N/A
3	0	0	N/A	0.3	0.305	N/A	0.71%	-0.1%	N/A
$g_1 = g_2 = g_3 = 0.5$ Centralized									
Node	Overshoot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	8%	2%	N/A	0.22	0.166	N/A	4%	0.01%	N/A
2	5%	0	N/A	0.22	0.314	N/A	0.1%	1.6%	N/A
3	0	0	N/A	0.225	0.32	N/A	0.02%	0.7%	N/A

Table 3: Performance results using different gains (con. 2)

$g_1 = 0.1, g_2 = 0.5, g_3 = 0.9$ Decentralized									
Node	Overshoot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	0	0	0	0.05	0.051	0.533	0.3%	2%	0
2	0	0	0	0.547	0.553	0.922	-0.8%	0.4%	-0.8%
3	0	0	N/A	1.455	1.454	N/A	-0.7%	0	N/A
$g_1 = 0.1, g_2 = 0.5, g_3 = 0.9$ Centralized									
Node	Overshoot			Settling time			Steady status error		
	P	O	B	P	O	B	P	O	B
1	8%	0	0	1.01	0.52	0.6	4%	2%	0
2	4%	0	0	1.02	0.56	1.065	0.02%	0.4%	-0.75%
3	0	0	N/A	1.02	1.48	N/A	0	0.1%	N/A

From the above three tables, we find that the greater the forward gains (g_1, g_2), the faster the system reaches its equilibrium status (settling time). We also notice that the decentralized method gets less overshoot than the centralized one, and has less settling time. When centralized method is used, the settling time of the premium traffic of each node is almost simultaneous. This is because the centralized algorithm needs all the information of three nodes to evaluate, so the queue length of the premium traffic in each node will get its equilibrium status simultaneously. However, for the decentralized method, it evaluates independently, so the queue length will achieve its equilibrium separately. For the first node, generally the centralized method has much more steady state error than the decentralized one; while for the second and third nodes, it has less

error.

For large gains, for example, if we choose $g_1 = g_2 = g_3 = 0.9$, under this situation, we cannot get good results for both methods. Figures 4-78 and 4-79 show the queue length of the premium traffic of each node based on different methods. We notice that for node 1 the buffer size is keep increasing, which is obviously unstable.

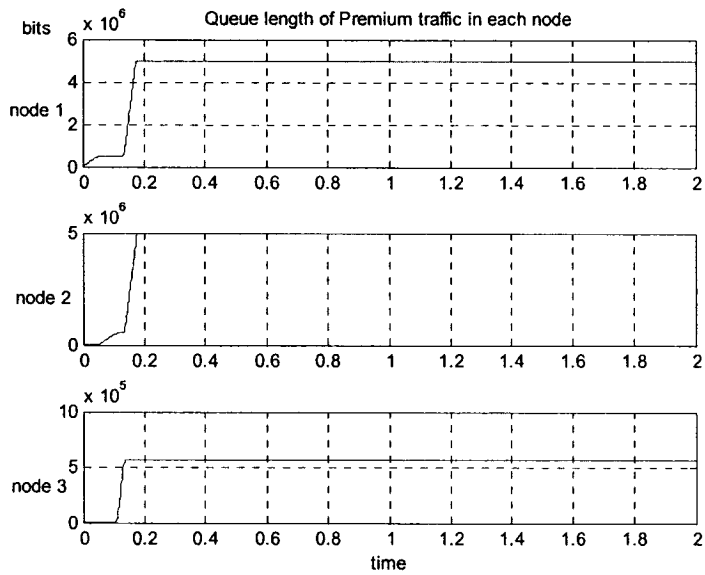

Figure 4-78: Queue length of premium traffic under decentralized method

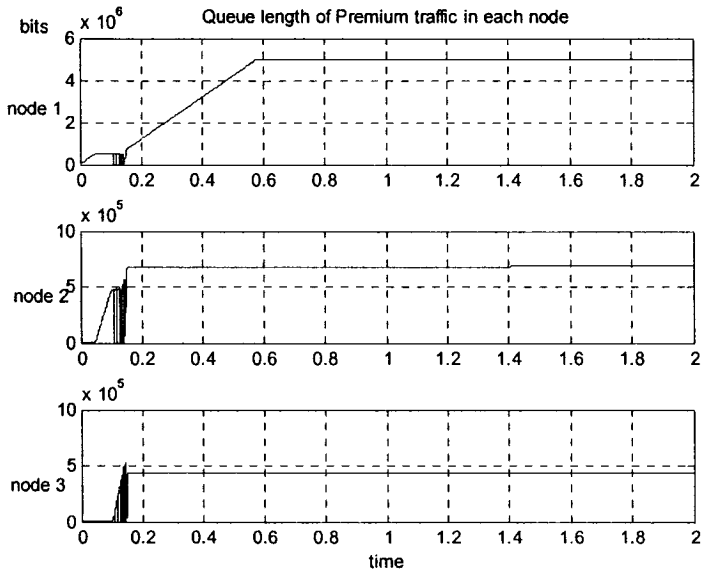

Figure 4-79: Queue length of premium traffic under centralized method

4.4.3 Summary

Based on the results in this chapter, we can state that the decentralized method gets less overshoot than the centralized one, and has less settling time. For the first node the centralized method has much more steady state error than the decentralized one; while for the second and third nodes, it has less error. Although the centralized algorithm can tolerate more delay, the decentralized method is much easier to implement.

Chapter 5

Contributions and Future Work

5.1 Contributions

Recently congestion control algorithms based on fluid flow model and non-linear control theory have been widely studied. Our objective is to efficiently use the finite link capacity while maintaining well controlled system performance, including stability, rise time, overshoot and other design criteria. In this thesis, a simple dynamic congestion control approach is introduced in Chapter 3, and we have improved this algorithm to apply to a switching control method. This method can choose from the two different control algorithms according to the traffic flow rate. For example, when the incoming traffic of ordinary traffic is insufficient, we will calculate the allocated bandwidth of it instead of calculating the allowed rate which will occupy all the leftover bandwidth [16], [45].

Due to the increasing need and requirements, the topology and structure of the network become more and more complex, and more feedback information is needed to monitor the whole network's performance. Such a simple structure for a three-node network is

shown in Figure 1-5 which simulates the Sensor-Decision Maker-Actuator structure. Feedback information is from decision maker to sensor and from actuator to decision maker and to sensor. Based on this structure, we designed centralized and decentralized methods to regulate the traffic flow with three different priorities (premium, ordinary and best effort) among three nodes (Sensor – Decision maker - Actuator), and analyzed the stability of the system.

We have simulated these models by Simulink and Matlab under different scenarios to show that they can offer satisfactory performance for the control system designs. For instance, we use different delay and gain to investigate the dynamic ability of those systems. The results from the dynamic fluid flow model show us that the behaviors of our model is very close to an event based real world network environment. Furthermore, we have compared the performance of both centralized and decentralized methods. For example, we investigated how the delays and gains will affect the system's performance under the two control methods, and also summarized the comparison results.

5.2 Future Directions

In part, due to lack of structured approach, and lack of strong theoretical foundation in networked control systems, most proposed schemes are developed using intuition and simple non-linear designs. When using simulation tools, these simple schemes demonstrated to be robust in variety of scenarios. However, problem is that very little known why these methods work and very little explanation can be given when they fail.

It is shown that the models that use non-linear control theory are suitable for congestion control of a simple networked system. The simulation results discussed in Chapters 3 and 4 encourage us to investigate this model further to see if it can be proposed as a general congestion control scheme suitable for more general networks. As shown in Figure 1-2, in a real network, every node is fully connected and the structure of the network is very complex. There is much more than three incoming traffic to each node. Additionally, the problems of delay, communication and routing should also be considered, so 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. IP Diff-Serv)
- Extension to large scale networks
- Extensive simulations and evaluation of results

Appendix A

If a 3x3 Matrix P ($p_{ij}(i,j=1,2,3)$ is the element in Matrix P) is stable, then the following four inequalities must be satisfied.

$$eq1 = -(p_{11} + p_{22} + p_{33}) > 0$$

$$eq2 = p_{11}p_{22} + p_{11}p_{33} + p_{22}p_{33} - p_{13}p_{31} - p_{32}p_{23} - p_{12}p_{21} > 0$$

$$eq3 = p_{11}p_{23}p_{32} + p_{22}p_{13}p_{31} + p_{33}p_{12}p_{21} - p_{11}p_{22}p_{33} > 0$$

$$eq4 = eq1 * eq2 - eq3 > 0$$

Proof:

Note that

$$|sI - P| = \left\| \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \right\| = \left\| \begin{bmatrix} s - p_{11} & -p_{12} & -p_{13} \\ -p_{21} & s - p_{22} & -p_{23} \\ -p_{31} & -p_{32} & s - p_{33} \end{bmatrix} \right\|$$

$$= s^3 - (p_{11} + p_{22} + p_{33})s^2 + (p_{11}p_{22} + p_{11}p_{33} + p_{22}p_{33} - p_{13}p_{31} - p_{32}p_{23} - p_{12}p_{21})s \quad (*)$$
$$+ (p_{11}p_{23}p_{32} + p_{22}p_{13}p_{31} + p_{33}p_{12}p_{21} - p_{11}p_{22}p_{33})$$

According to the Routh-Hurwitz stability criterion, for a third order characteristic function

$s^3 + B_2 s^2 + B_1 s + B_0 = 0$, the necessary and sufficient conditions that all roots are located in the left-half phase are:

$$B_2 > 0; B_1 > 0; B_0 > 0; B_1 B_2 - B_0 > 0;$$

Compared with (*), we can easily find that matrix P is stable, if

$$eq1 = -(p_{11} + p_{22} + p_{33}) > 0$$

$$eq2 = p_{11}p_{22} + p_{11}p_{33} + p_{22}p_{33} - p_{13}p_{31} - p_{32}p_{23} - p_{12}p_{21} > 0$$

$$eq3 = p_{11}p_{23}p_{32} + p_{22}p_{13}p_{31} + p_{33}p_{12}p_{21} - p_{11}p_{22}p_{33} > 0$$

$$eq4 = eq1 * eq2 - eq3 > 0$$

Reference

[1] G.C. Walsh and Y. Hong and L.G. Bushnell. "Stability, Analysis of Networked Control Systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp.438-446, May 2002.

[2] J.K. Yook, M.Tilbury, and N.R. Soparkar. "Trading Computation for Bandwidth: Reducing Communication in Distributed Control Systems Using State Estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, pp.505-508, July 2002.

[3] W. Zhang and M.S. Branicky and S.M. Phillips. "Stability of Networked Control Systems," *IEEE Control System Magazine*, vol. 21, pp. 84-99, February 2001.

[4] A Makarenko, A Brooks, S Williams, H Durrant-Whyte, B Grocholsky, "A Decentralized Architecture for Active Sensor Networks," IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, April 2004.

[5] Feng-Li Lian, "Analysis, design, modeling and control of networked control systems," Ph.D. Dissertation, Dept. of Mechanical Engineering, The University of Michigan,2001

[6] Michael S. Branicky, Vincenzo Liberatore, and Stephen M. Phillips, "Networked control system co-simulation for co-design," *Proc. American Control Conf.*, Denver, June 2003.

[7] C. F. Reverte and P. Narasimhan, "Decentralized Resource Management and Fault Tolerance for Distributed CORBA Applications," *IEEE Workshop on Object-oriented Real-time Dependable Systems*, Capri Island, Italy, October 2003

[8] L.H.Eccles. A smart sensor bus for data acquisition. Sensors, 15(3):28-36,1998.

[9] N.Najafi and J.R. Moyne. Smart sensors. Semiconductor micromachining. S.A. Campbell

and H.J.Lewerenz. editors. John Wiley & Sons, 1998.

[10] A. Kawarnura, J. Hirai, and Y. Aoyama. Autonomous decentralized manufacturing system using high-speed network with inductive transmission of data and power. In proceedings of the IEEE 22nd International Conference on Industrial Electronics, Control, and Instrumentation, volume 2, pages 940-9435, Aug. 1996.

[11] ITU-T Recommendation I.371, Traffic Control and Congestion Control in B-ISDN, March 1993.

[12] V. Jacobson, "Congestion Avoidance and Control," Proceedings of SIGCOMM '88, Palo Alto, CA, USA, August 1988.

[13] D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks," Journal of Computer Networks and ISDN, Vol. 17, No. 1, pp. 1-14, June 1989,

[14] S. Keshav, "A control theoretic approach to flow control," ACM SIGCOMM'91, Zurich, Switzerland, 1991

[15] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.

[16] A. Pitsillides, P. Ioannou, L. Rossides, "Congestion Control for Differentiated Services using Non-Linear Control Theory," in Proceedings of the Sixth IEEE Symposium on Computers & Communications, ISCC 2001, Hammamet, Tynisia, pp. 726-733, July 2001

[17] P.-F. Quet, B. Ataslar, A. Iftar, H. AOzbay, T. Kang and S. Kalyanaraman, Rate- based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays," Automatica, 38(6):917-928, June 2002.

[18] Rajesh S. Pazhyannur, Rajeev Agrawal, Rate- based flow control with delayed feedback in integrated services Networks," Technical Report ECE-97-4, ECE Dept., University of Wisconsin-Madison, USA, July 1997.

[19] J-C. Bolot, A.U.Shankar, "Analysis of a fluid approximation to flow control dynamics," Proc.

IEEE Infocom 1992, pp. 2398-2407, Florence, Italy, May 1992.

[20] 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, pp. 785-793. March, 1996.

[21] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, January 1997.

[22] L. Roberts, "Enhanced PRCA (Proportional rate control algorithm)", Tech. Rep. AF-TM 94-0735R1, August 1994.

[23] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, ERICA switch algorithm; a complete description, ATM FORUM, AF/96-1172, August 1996.

[24] A. Segall, "The modelling of adaptive routing in data communication networks", IEEE Transactions on Communications, Vol. 25, No.1, pp. 85-95, January 1977.

[25] 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.

[26] L. Benmohamed, Y.T. Yang, "A Control-Theoretic ABR Explicit Rate Algorithm for ATM Switches with Per-VC Queuing", Infocom 98, 1998.

[27] 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.

[28] 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.

[29] 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.

[30] A. Pitsillides, A. Sekercioglu, Fuzzy Logic based effective congestion control, TCD

workshop on Applications of Computational Intelligence to Telecommunications, London, 16 May, 1999.

[31] 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.

[32] 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.

[33] 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.

[34] A. Isidori, "Nonlinear Control Systems", 2nd Ed. New York, Springler-Verlag, 1989.

[35] 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.

[36] Z. Arstein, "Stabilization with relaxed controls", Nonlinear Anal., 1983, pp. 1163-1173.

[37] 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.

[38] M. Krstic, I. Kanallakopoulos, P.V. Kokotovic, "Nonlinear and Adaptive Control Design", John Wiley&Sons, 1995.

[39] K.S. Narendra, S. Mukhopadhaay, "Adaptive control using neural networks and approximate models", IEEE Transactions On Neural Networks, vol. 8, 1997, pp. 475-485.

[40] 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.

[41] 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.

[42] P.A. Ioannou and J. Sun, Stable and robust Adaptive Control, Englewood Cliffs, NJ: Prentice Hall, 1996.

[43] 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.

[44] 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.

[45] A. Pitsillides, A. Sekercioglu, Congestion Control, in Computational Intelligence in Telecommunications Networks, (Ed. W. Pedrycz, A. V. Vasilakos), CRC Press, ISBN: 0-8493-1075-X, September 2000, pp- 109-158.

[46] Loukas Rossides, Congestion Control in Integrated Services Networks, M.A.Sc Thesis, Department of Computer Science, University of Cyprus, December 1999.

[47] George Hadjipollas, Simulative Evaluation of IDCC in a Resource Management Differentiated Services Environment, M.A.Sc. Thesis, Department of Computer Science, University of Cyprus, June 2003.

[48] D. Tipper, M. K. Sundareshan, "Numerical Methods for modeling Computer Networks Under Nonstationary Conditions", IEEE JSAC, Dec. 1990.

[49] J. Filipiak, "Modelling and Control of Dynamic Flows in Communication Networks", Springer-Verlag, 1988.

[50] S. Sharma, D. Tipper, "Approximate models for the study of nonstationary queues and their application to communication networks", IEEE ICC'93, May 1993.