Modeling and Verifying a Price Model for Congestion Control in Computer Networks Using PROMELA/SPIN

Clement Yuen and Wei Tjioe

Department of Computer Science
University of Toronto
10 King's College Road,
Toronto, Ontario M5H 3G4, Canada
{clhyuen,wtjioe}@cs.toronto.edu

Abstract. Congestion control is an important research area in computer networks. Using PROMELA/SPIN, we verified that priority pricing schemes [1-4] can be used to effectively control network congestion. This is realized through simulation/verification of the propositions that the use of priority pricing (i) results in an equilibrium state in packet allocation, and (ii) effectively controls congestion level through dynamic adjustment of prices. We also extended these propositions in order to verify the convergence property of such an equilibrium. This particular result would be difficult to verify with existing network simulation tools.

1 Introduction

PROMELA/SPIN is a versatile tool in the simulation and verification of software systems [5]. A common application of PROMELA/SPIN is in modelling and verifying communication protocols [6]. In the area of mobile communication, part of the MCNet architecture has been simulated and verified in SPIN [7]. With Java PathFinder [8], it is now possible to translate programs written in JAVA version 1.0 to PROMELA. In addition, PROMELA/SPIN has been showed to be useful in modeling business rules [9]. Inspire by these novel applications, we examine the applicability of SPIN for a nontrivial pricing model in relation to congestion control in computer networks.

As demand for Internet services increases, so is the importance of congestion control in network traffic. Conventionally, congestion control can be accomplished by (i) sending control packets from an overloaded node to some or all of its sources, (ii) providing delay information in influencing network routing decisions, or (iii) making use of end-to-end probe packets and time stamps to measure the delay between two end nodes. One drawback from these approaches is that more control packets must be added to the network before data traffic can be reduced.

Putting network users in a more active decision making position is a novel idea emerged from recent congestion control researches [2,3]. Priority pricing scheme is one implementation of this idea. The quality of packet transmission is associated with the price a user is willing to pay. The establishment of prioritized service class based on pricing will enable network provider to adjust network volume through pricing. However, ideas based on priority pricing remain difficult to verify due to the complexity of their scopes. We discuss two such ideas below.

Gupta [4] presented a robust model using priority pricing in the dynamic management of network bandwidth. In this model, a network node may return an estimated price and waiting time to a user in response to the quality of service requested. After collecting the current pricing and predicted waiting times from all servers, the user calculates the total expected cost in using the network. If the estimated cost is higher than the benefit of using the network, the user may decide to delay transmission until a more agreeable price becomes available. Otherwise, the user may begin releasing packets into the network. Note that the cost and waiting time may increase when traffic volume becomes high. In response to increased cost, the user will reduce the volume of packets released into the network. However, network delays are common and random enough that updated pricing information may not reach the user in time to become useful. Therefore, this pricing scheme can be difficult to implement and verify.

Marbach [1] described a simpler model that may become more effective in practice. In this model, each priority transmission level is associated with a “fixed” price (on a fast time-scale). Among this hierarchy of pricing choices, the pricing at two consecutive levels are of great significance. In this paper, we will refer to the higher price as the “premium priority price”, and the lower one the “best effort price”. When packets are released into the network with premium priority, they are assured to be transmitted successfully. Packets released with best-effort priority will always be associated with a certain non-zero probability of packet drop. All submissions at a price higher than the premium are guaranteed to succeed, while submissions at a price below the best-effort priority are bound to be dropped. Users are expected to dynamically adjust the quantity of packets submitted (on a fast time-scale) to maximize their economic benefits. Marbach’s model showed that an equilibrium in packet allocation from network users can be established under priority pricing scheme. This equilibrium can potentially be manipulated to control congestion.

While there are many excellent tools available for network simulation [10-12], they do not provide any verification capability. When a model is introduced, its correctness must be analytically proven before it can be used as the bases for further discussion. Simulation is an alternative to a formal proof when analytical verification becomes too difficult. However, simulation cannot provide as strong an assertion on any important results. In this regard, PROMELA/SPIN is a valuable alternative in asserting properties that may otherwise be difficult to verify.

In this work, we will examine the applicability of PROMELA/SPIN as a simulation-verification tool for nontrivial congestion control models based on Marbach’s priority pricing model [1]. Our objective is to examine whether SPIN can be used to provide effective assertion to important properties of the underlying model, and demonstrate the stronger results bring forth through SPIN’s unique simulation-verification framework. In particular, we will apply SPIN to simulate exhaustively the correctness of Marbach’s priority pricing proposition and provide verification in LTL on its convergence property. Furthermore, we will extend this model with a SPIN simulation to show that allocation equilibrium can be maintained under the dynamic pricing scheme suggested by Gupta [4]. This latter result has not been analytically proven and would be very hard to verifying with existing tools in network simulation. We have chosen SPIN for this work due to its inherent power and flexibility in modeling complex software systems. We would like to examine the possibility of expanding the scope of SPIN for models based on economic or mathematical frameworks.

The remainder of this paper is organized as follows. Section 2 provides an overview of the price model, along with two proposed models for verification of its properties. Section 3 presents the design details and architecture of our models. Section 4 discusses details of simulation and verification in PROMELA/SPIN. Section 5 discusses results and implementation limitations. We conclude this paper in Section 6.

2 Overall Design of the Pricing Model

This section introduces Marbach’s priority pricing model with highlights on the pertinent details and terminologies that is fundamental to discussions throughout this paper. Propositions of Marbach’s model are then presented. We designed two models with PROMELA/SPIN to verify Marbach’s propositions. Model 1 verifies Marbach’s proposition that an equilibrium in packet allocation exists. Furthermore, it verifies the convergence of this equilibrium. Model 2 is a natural extension to Model 1. It contains an additional process in manipulating transmission prices. In effect, this process allows us to verify whether congestion control can be achieved by dynamically adjusting priority pricing.

2.1 The Mathematical Priority Pricing Model

A computer network that implements priority pricing is modelled in [1] based on the following framework:

- A discrete-time, single-link bufferless *channel* with a fixed capacity **C**.
- There are *R users* sharing the channel, **R** = {1, …, *R*}.

- Network services are provided in N different *priority classes*, $\mathbf{N} = \{1, \ldots, N\}$. Class i is at a higher priority than class j if $i > j$, $i, j \in \mathbf{N}$. Packet traffic of a higher priority class receives preferential treatment over a lower priority class.
- Associated with each user $r \in \mathbf{R}$ are:
  - allocation quantities denoted by $d_r(i)$ which are the amount of packets allocated by the user at priority $i \in \mathbf{N}$ in each time slot; and
  - a private *utility function* U_r known to the user. A utility function is an economics concept that reflects a user's monetary gain as a result of using the network. It is assumed to be a strictly increasing function of the perceived throughput, denoted by x_r.
- Associated with each priority class $i \in \mathbf{N}$ are:
  - a *transmission probability* $P_{tr}(i)$ defined as

$$P_{tr}(i) = \begin{cases} 1 & \text{if } \mathbf{C} > \sum_{k=i}^{N} d(k) \\ \frac{\mathbf{C} - \sum_{k=i+1}^{N} d(k)}{d(i)} & \text{if } \sum_{k=i}^{N} d(k) \geq \mathbf{C} > \sum_{k=i+1}^{N} d(k) \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

  where $d(i) = \sum_r d_r(i)$ is the aggregated allocation of packets over all users at priority i. This is the probability that packets allocated (submitted) at priority i are successfully transmitted; and
  - a *price* u_i . In a priority pricing scheme, $0 < u_i < u_j$ whenever $i < j$, $i, j \in \mathbf{N}$. Submitting packets at a higher priority is more expensive.

It can be seen from the definition of transmission probability (1) that a *best-effort* class i_0 exists such that (a) $0 < P_{tr}(i_0) \leq 1$, (b) $P_{tr}(i) = 1$ for all $i > i_0$ and (c) $P_{tr}(i) = 0$ for all $i < i_0$. In other words, packets in best-effort class are transmitted with a non-zero probability; packets from higher priority classes are guaranteed to be transmitted; packets from lower priority classes are all dropped. In this paper the class i_0+1 is referred to as the *premium* class. u_{i_0} is referred to as the *best-effort price*, and u_{i_0+1} the *premium price*.

In a discrete-time formulation, the following interactions take place in every time slot. The network channel provides priority services according to the probability distribution in (1). Network user determines an optimal allocation of packets (i.e. $d_r(i)$) for the next time slot according to the same transmission probabilities. The optimality arises from the fact that the quantity of packets allocated from the network maximizes the net economic gain of the user. In mathematical terms, the determined allocation for user r is the solution of the maximization problem $\max_{d_r(i) \geq 0} U_r(x_r) - \sum_i d_r(i) \cdot u_i$. Let $G_r(x_r)$ (the marginal utility function) be the derivative of $U_r(x_r)$, and G_r^{-1} its inverse function, then it can be shown analytically that when best-effort cost is lower than premium cost, allocation is optimal with $d_r(i_0) = \frac{1}{P_{tr}(i_0)} G_r^{-1}(u_{i_0} / P_{tr}(i_0))$, $d_r(i) = 0$ for all $i \neq i_0$. This means all packets should be submitted to the best-effort class to perceive the best benefit. When best-effort cost is higher than premium cost, allocation is optimal with $d_r(i_0+1) = G_r^{-1}(u_{i_0+1})$, $d_r(i) = 0$ for all $i \neq i_0 + 1$. This means all packets should be submitted to the premium class to perceive the best benefit. Finally, when best-effort cost equals premium cost, user receives identical benefit from submitting packets to either class. Hence both allocation strategies are optimal. A network user makes use of the above strategy in determine their allocation of packets for the next time slot.

Under this model, Marbach [1] provided an analytically proof for the following two propositions.

Propositions:

I. Under a fixed pricing scheme, an *equilibrium* condition exists in packet allocations of channel users.
II. Priority pricing is effective in congestion control.

At equilibrium, each user's allocation maximizes its own net benefit. Therefore, no user has an incentive to change its allocation. Therefore *equilibrium* is defined to be the state where no user allocation will deviate from its current value. In the model, this condition can be realized when all users' allocations remain unchanged between any two consecutive time slots.

Note that congestion level of the network channel can be reflected in the sum (aggregate) of packets allocated by all users. With a fixed transmission capacity, the channel is considered congested when packet submission exceeds its capacity. This in reality will result in packet drops and poor Quality of Service. Congestion control is observed through proposition II.

In order to verify propositions I and II, we formulate the following models in PROMELA/SPIN and show their relationships with Marbach's analytical model [1] in terms of their main components and objective of their propositions.

2.2 Model 1 (Demand Equilibrium Model)

Model 1 is designed to capture the major ingredients of the mathematical model. It is readily translated into PROMELA and verified in SPIN. It also serves as the basis for Model 2 where congestion control is verified. Model 1 is made up of the following main constituents.

- A channel process with a fixed capacity
- R user processes running in parallel with the channel process (sharing the channel).

A user process can probe the channel for the transmission probabilities on a frequent but fair basis. It derives an optimal allocation of packets for the next time slot using the strategy described in Section 2.1. A channel process computes the transmission probabilities of the network according to (1). This value is made available to all users of the network.

The objective of Model 1 is to examine the validity of proposition I through the simulation and verification framework provided by PROMELA/SPIN. Furthermore, Proposition I is enhanced to a stronger postulation which we now restate as proposition 1.

Proposition 1

Under a fixed pricing scheme, an equilibrium condition in packet allocations of channel users exists and *converges* from some initial allocations.

Since Promela/SPIN only work with discrete values, an equilibrium condition in Model 1 is considered attained when the differences between consecutive allocations of all users are bounded by a small integer. In this model we choose the bound to be 1.

It should be noted in general that convergence properties are difficult to prove analytically. In particular an analytical proof for convergence in proposition 1 has not been given in [1]. However with an automated verification tool such as SPIN, we are able to mechanically verify the correctness of such proposition.

2.3 Model 2 (Dynamic Price Adjustment Model)

In order to explore the effect of dynamic price adjustments and verify proposition 2, we extend Model 1 to Model 2 by adding an administrator process to the system.

The channel and user processes operate and interact in the same manner as in Model 1. The administrator process monitors the aggregated packet allocation at regular intervals and adjusts the best-effort price. In response to a price change, user processes adopt the same allocation strategy as described in Section 2.1 based on the new price given by the administrator.

The objective of Model 2 is to examine the validity of proposition II using PROMELA/SPIN. Proposition II is restated as Proposition 2 to more specifically address the mechanism in which priority pricing controls congestion.

> ***Proposition 2***
>
> Priority pricing can effectively change the aggregated equilibrium level of packet allocation.

3 Design and Architecture

This section describes the requirements, overall architecture, and various components that are important to the models.

The following requirement applies to both Models.

- The time-scale on which simulation is performed should be discrete. Users are expected to closely monitor the level of congestion as reflect by the transmission probability of the network, and react promptly by computing a new quantity of packets to submit for the next time slot.
- The channel process is expected to return up-to-date transmission statistics to its users in an equally timely fashion. As a result, both user and channel processes should proceed on a fast time-scale.
- Pricing for each priority classes should be maintained at a steady level to allow the system sufficient time to reaching equilibrium (if exists). Since this falls under the administrator's responsibility, the administrator process should proceed on a slow time-scale.

The main implementation choices applicable to Models 1 and 2 are:

- The channel process should be invoked following every change in user allocation.
- A user process should proceed immediately after each update of the network transmission probabilities.
- In order to guarantee that users submit packets at a constant and deterministic rate, a particular user should be required to wait (block) until all other users have updated their allocations for the next time slot. This mechanism is necessary to enforce strong fairness.

An additional implementation choice for Model 2 is:

- An administrator process should be invoked on occasions and dynamically adjust the best-effort transmission price if the network channel is currently in the state of equilibrium.

The overall architecture of Model 1 contains an environment where both user and the network interact dynamically. The successful operation of Model 1 requires user processes to submit packets to the channel process, and the channel process return the current transmission probability of the network to all users. Figure 1 describes the overall architecture of the environment containing user and channel processes.

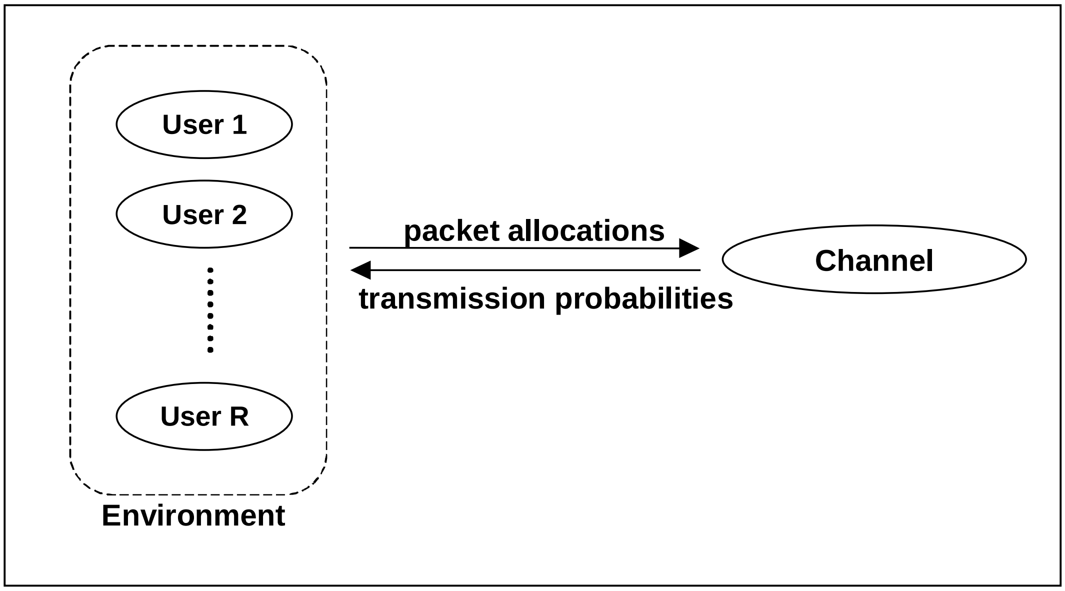

Fig. 1. Overview of Model 1 Architecture

The architecture of Model 2 contains a channel, an administrator, and user processes. In this architecture, the network channel communicates dynamically with both users and the administrator. Given the quantity of packets submitted by users, the channel process returns the current transmission probabilities to users. The administrator process communicates current pricing information to users in fixed but larger time intervals. Figure 2 describes the overall architecture of the environment containing user, administrator and the channel processes.

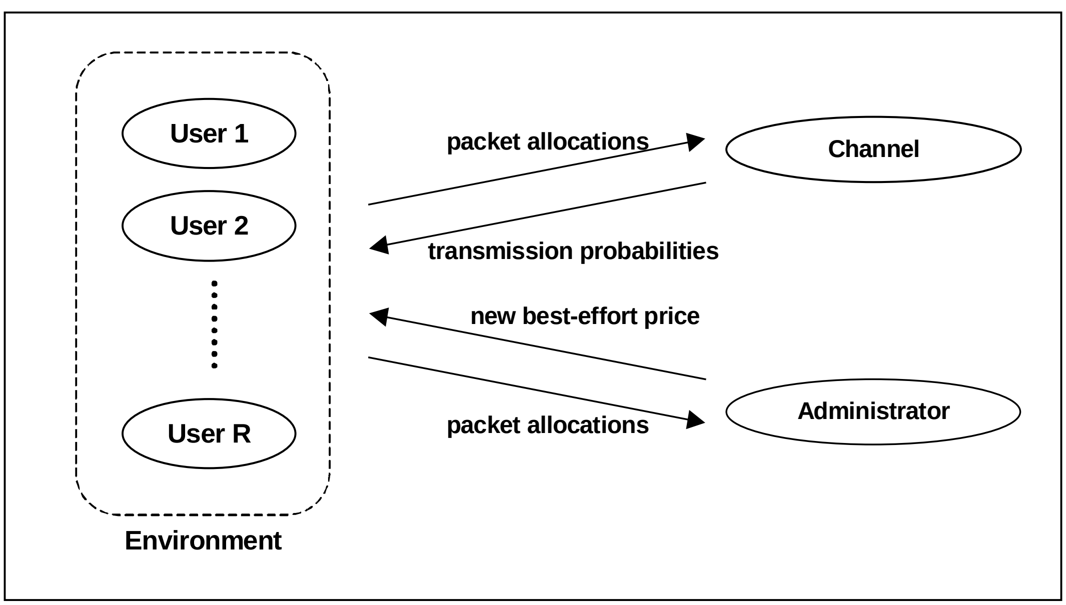

Fig. 2. Overview of Model 2 Architecture

Other important parameters in the modeling environment consist of the quantity of packets user submits, the transmission probability of the network channel, and an array of prices each associating with a priority class. The concept of priority pricing is represented as a hierarchy of pricing choices, with higher pricing values associated with increased transmission probability.

The condition in which equilibrium can be detected is also of great importance. It was not clear whether the limitation of PROMELA/SPIN as a discrete-value modeling tool will impact the outcome of simulation. In particular, it is unlikely to observe convergence in packet allocation to one single value for all cases. On the other hand, packet allocation may also diverge. Through experimentation, we found in general that the model did converge to within a very small range. Though the actual behavior depended on scalable parameters such as the number of users. An effective way to verify convergence within a small range is to defined in PROMELA a statement "`#define BOUND(A,B,C)  (A - B <= C) && (A - B >= -C)`". It maintains a previous (`B`) and current (`A`) value of interest and compares their distance with the bound (`C`).

Another area requiring special attention is the representation of rational numbers, or fractions in modeling. It is obvious that fractions such as transmission probabilities cannot be stored in integer variables as they are likely to be truncated. As a result, a `fraction` data type is devised to store rational numbers as pairs of integers (numerator, denominator). In this way, fraction multiplication can be translated to integer multiplication and division.

The quantity of users in modeling also requires some consideration. In general, modeling larger number of users provide better approximation to reality. When a network is shared by a very small amount of users, the model may not converge to within a reasonably small range. It may oscillate between two distant values. This result is inferred from the non-cooperative game assumption that a large number of users is required to buffer individual impact.

A system effectively modeling channel and user strategies should contain the following.

- A parameter N representing the number of priority classes.
- Parameters associated with each user process:
  - a data structure of allocation quantities, representing packet allocations to each class; and
  - a utility function defined for the user. This should be a simple function with the desired (strictly increasing) property.
- A data structure representing rational transmission probabilities.
- A data structure representing prices associated with priority classes.

3.1 Channel and User Strategies (Model 1)

The channel process should review the transmission probability of a network link in a timely fashion. It coordinates the execution of user processes and continuously remains active throughout the simulation. Since a communication channel usually assumes full capacity when it first becomes available, the transmission probability is initialized to 1. The channel process is also the first to be instantiated. This ensures that a communication infrastructure is available before user processes can possibly interact. The channel process proceeds by computing the available transmission capacity of the network link and determines the sum of total demands from users of the network. If the network capacity exceeds the sum of user demands, then the transmission probability of the network is set to 1. Otherwise, it is equal to available capacity of the link divided by the sum of total user demands. Note that the transmission probability is updated at each process cycle, and stored as a global variable. This arrangement effectively simulates the situation where users of a network can probe the network link for current congestion information.

The user process computes the next allocation of packets according to the current congestion information. The quantity of packet allocation is determined by the benefit each individual user receives from using the network. User benefit can be represented in terms of a utility function $U_r(x_r) = A_r\sqrt{x_r}$. This definition satisfies the general requirements that a utility function increases with strictly decreasing marginal value. The scaling constant A_r is assumed to be unique in relation to a user's financial strategy. For simplicity, we assume a homogeneous user population, adopting to the same scaling constant A_r.

In general, user strategy in packet allocation follows the derivation in Section 2.1. A user process is enabled only when transmission probabilities have been update by the channel process. All users in the network are guaranteed a fair chance in allocating transmission resource. In particular, 0 packets are allocated to all low priority classes up to but not including the best-effort class. If the condition ($u_{best-effort} / P_{best-effort} \geq u_{premium}$) is true, then user submits $A^2 / 4u^2_{premium}$ packets at premium class. When condition ($u_{best-effort} / P_{best-effort} \leq u_{premium}$) becomes true, user submits $A^2 P_{best-effort} / 4u^2_{best-effort}$ packets at best-effort class for transmission. Zero quantity is allotted to all remaining high priority classes. Note that when both conditions 1 and 2 becomes true, the user process nondeterministically selects either class to allot.

3.2 Administrator Process (Model 2)

The object of administrator process is to introduce pricing perturbations into the system at designated time intervals. It calculates the sum of current user demand and determines if equilibrium has been established. It then asserts whether the total demand and transmission price at equilibrium exhibits an inversely proportional relationship. When equilibrium is asserted, the administrator process non-deterministically picks the new best-effort price from the set { $u_{init} - 1$, u_{init}, $u_{init} + 1$ } where u_{init} represents the initial value for the best-effort class. Users are forced to re-calculate their economic benefit under the new price, and adapt a packet allocation that will optimize their benefits in the next process cycle. As a result, the transmission probability of the network is dynamically changed at run time.

4 Simulation and Verification

Simulations and verifications were performed in order to observe convergence in packet allocation, and its stability thereafter. Since Model 1 and 2 were designed to mimic the continuous operation of a network channel, each simulation would not normally terminate. There is really no final state in both Models 1 and 2. Data was therefore collected through simulations of 5 seconds trials. Approximately 1500 data point can be obtained from each trial. Simulation and data collection for both Models 1 and 2 followed the same procedure, however, each simulation was conducted independently of each other. For data analysis, simulation of Model 1 was executed three times, and the averaged data set was taken in the construction of Figure 3. Simulation of Model 2 was performed numerous times, however only data from one trial is shown in Figure 4. Each simulation of Model 2 mimics a self-contained network system, with an administrator arbitrarily adjusting prices. Therefore, there is little coherence between different executions of the model to justify averaging of data.

All simulations and verifications were performed using SPIN Version 3.4.3 on a Sun UltraSPARC server with 4 400MHz processors and 4GB of physical memory.

4.1 Simulation of Channel, User and Administrator Processes

Table 1 tabulates the simulation of Model 1. It shows the data value from the first 20 cycles of the interaction between the user process and the channel process. In particular, rows 1 to 10 show user packet allocations from the first ten cycles. Rows 11 to 20 show subsequent packet allocations for the same users. At Time 1, User 7 allocates 2500 packets under priority class 1. The total aggregated demand for the entire channel at this time is only 2500 packets. Other users have not yet allotted their submissions. At Time 2, User 0 allotted 2000 packets to the network. The aggregate demand at this instance becomes the sum of allotments of both User 7 and User 0. Before User 3 releases its submission to the network at Time 3, it has computed that submitting at priority class 2 carries more advantage than a lower class. Therefore, a shift in allotment to higher priority classes can be observed in subsequent cycles. After all 10 users have released their submissions to the network, the allocation cycle repeats again. The order in which users release their packets to the network in the new cycle is determined randomly. This feature is important in simulating a working network where the arrivals of packets are commonly considered as random and independent events.

The aggregated demand represents the current allocation of all users of the network. For example, at Time 1, User 7 allocates 2500 packets for the first time slot. At Time 13, User 7 updates its allocation to 277 packets. Hence, at Time 13, `aggregated demand = aggregated demand + 227 – 2500.`

		Priority N					
Time	User	1	2	3	4	5	Aggregated Demand
1	7	2500	0	0	0	0	2500
2	0	2000	0	0	0	0	4500
3	4	0	625	0	0	0	5125
4	9	0	625	0	0	0	5750
5	2	0	625	0	0	0	6375
6	1	0	625	0	0	0	7000
7	5	0	500	0	0	0	7500
8	6	0	416	0	0	0	7916
9	8	0	0	277	0	0	8193
10	3	0	0	277	0	0	8470
11	9	0	0	277	0	0	8122
12	8	0	0	277	0	0	8122
13	7	0	0	277	0	0	5899
14	2	0	0	277	0	0	5551
15	1	0	0	277	0	0	5203
16	3	0	0	277	0	0	5203
17	5	0	0	277	0	0	4980
18	0	0	0	277	0	0	3257
19	6	0	0	250	0	0	3091
20	4	0	0	225	0	0	2691

Table 1. Aggregated User Demand

The average aggregated demand in total packets allocation is illustrated in Figure 3.

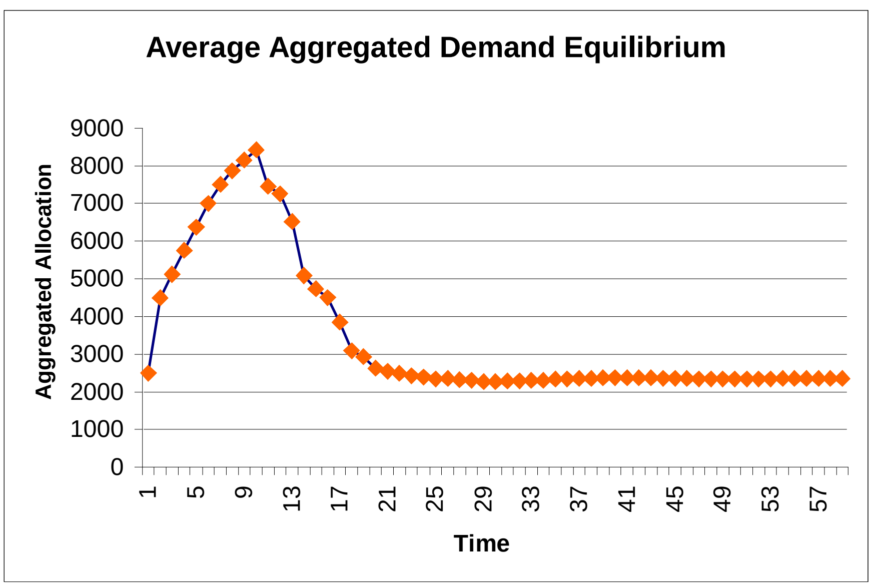

Fig. 3. Equilibrium in Pricing Model

In Figure 3, the aggregated allocation (demand) from all network users is presented on a fast time-scale. The data was averaged over three simulation runs. Before network becomes saturated, all users upload as many packets as desired under the assumption that transmission probability equals one. Guided by the solution of their benefit function, users realize that submitting as desired is not economically optimal. Total demand from users therefore begins to shift towards a value that would optimize everyone's benefit function. It reaches a plateau around iteration (time) 20 and remains relatively unchanged thereafter. This remarkable result indicates that the system has reached an equilibrium condition in packets allocation. Each user, in optimizing it's own economic benefit, participated in a non-cooperative game with the outcome of effectively drifting the network traffic to a steady level. This essentially supports Proposition I that an equilibrium is indeed possible under priority pricing scheme.

To simulate Model 2, an administrator process is added. Data values between every price change follow the trend observed in the simulation of Model 1. Model 2 can be considered as a sequence of Model 1 simulation, each with a different best-effort price maintained by the network administrator. Figure 4 illustrates the simulation of Model 2.

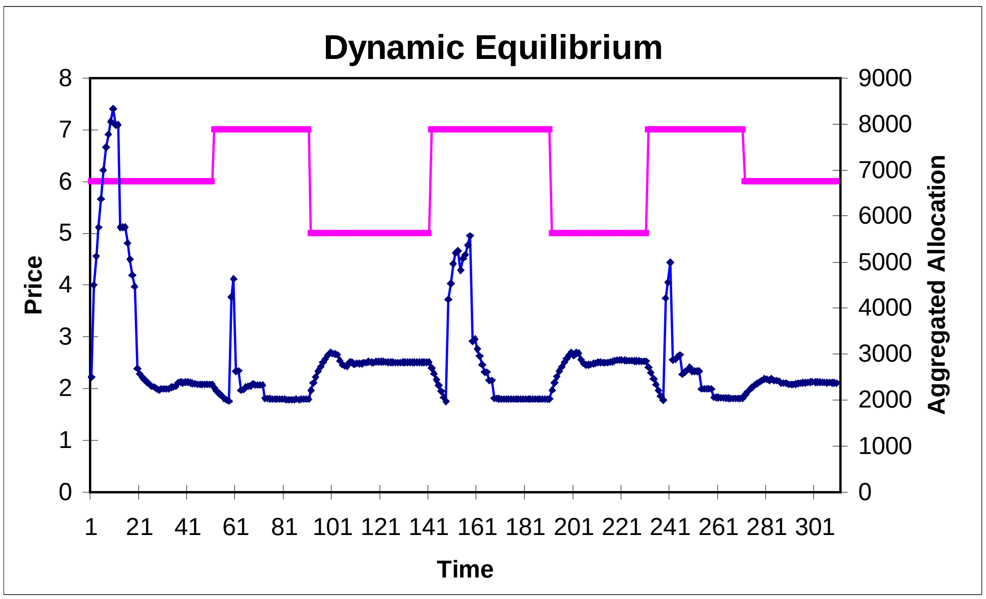

Fig. 4. Aggregated Equilibrium

The upper curve in Figure 4 represents the current best-effort price determined by the network administrator. The lower curve is the aggregated allocation for all users in the network along simulation time. To illustrate the effect of dynamic price adjustment, we can focus on the data segment between Time 100 and Time 200. The best-effort price at Time 100 starts with a dollar value 5. The aggregated allocation stabilizes to an equilibrium of about 2800 packet units at Time 114 and remains stable at this level. Note that the network capacity is only 2000 packet units. The probability of transmission is clearly below 1 at the current best-effort price. Seeing that network traffic is maintained at a steady level, the administrator process increases the best-effort price to 7. Facing a new higher cost in using the network, users begin to reduce the amount of packet allocation, resulting in a downward incline observed in the aggregated allocation curve. This phenomenon continues to Time 148, where the total allocation drops to 1975 packet units, which is below the total link capacity. As a result, a transmission probability of 1 is returned to the next user. This user allocates as many packets as desired under the false assumption that transmission for all packets submitted would be successful. Subsequent computations of the benefit function remind users that network bandwidth is not as abundant as anticipated. Each user then reduces its allocation to maximize the economical benefit under the new price. As a result, a new equilibrium at a lower aggregated allocation level is observed.

4.2 Verification of Convergence to Equilibrium (Proposition 1)

To verify that an equilibrium exists and converges, we formulate Proposition 1 as an LTL property and search for an acceptance cycle for the corresponding Büchi automata. Model 1 essentially describes the interleaving executions of the channel and user processes. The presence of an equilibrium entails the model's stability. It was for this reason that we opted to leave out the administrator process in this scenario so that the temporal nature of convergence could be verified.

We have formulated the following LTL property to specify this proposition for one user and submitted to SPIN:

$$\Diamond\Box\left(curr\_cl = prev\_cl\right)\wedge\left(d_0(curr\_cl)-d_0(prev\_cl)\leq 1\right)\wedge\left(d_0(curr\_cl)-d_0(prev\_cl)\geq -1\right)$$

This property indicates that "eventually user 0 will allocate its packets to a single priority class and that allocation will remain constant. Alternatively, user 0 has reached an equilibrium in its packet allocation". For R number of users in the network, there will be R such LTL properties one for each user. Due to homogeneity, we can anticipate consistent results from all users. Note that the LTL "always" operator (•) in the above verifies the existence of equilibrium in packet allocation while the LTL "eventually" operator (•) verifies the convergence of such equilibrium.

We configured SPIN to use the maximum amount of available memory, 4096MB, and allowed a search depth of 30,000,000. We also adopted 26-bit bit-state hashing in our verification. A typical run resulted in a 668-byte state vector, with 3.87348×10^7 states and 8.25047×10^7 transitions. In general verification of proposition 1 lasts for four hours with the given maximum search depth.

4.3 Verification of Dynamic Pricing Adjustment

With the addition of an administrator process, Proposition 2 can be verified by means of an assertion statement embedded into the PROMELA code. To verify this proposition, the administrator process is introduced to perform pricing adjustments on the best-effort class. Specifically we would like to derive and verify an inverse relationship between price and allocation at equilibrium. The following assert statement is embedded into the administrator process where equilibrium is detected:

assert(BOUND($d_{all}(curr\_cl)$ * Price($curr\_cl$),K,20));

This statement asserts that "the product of total demand (allocation) and price of the current allocated class of any user is equal to K"

We configured SPIN to use the same resources as in the case of proposition 1. A typical run resulted in a 692-byte state vector, with 3.93016×10^7 states and 8.03291×10^7 transitions. In general verification of proposition 2 lasts for two and a half hours with the given maximum search depth.

5 Discussion

This section briefly discusses the role of PROMELA/SPIN in reaching the objectives of Models 1 and 2. It also discusses some limitations we have experienced during the course of design and simulation.

5.1 Contribution of PROMELA/SPIN in Model 1

Model 1 was constructed based on the mathematical framework of Marbach's Model [1]. It is distinguished from Marbach's model with the verification that convergence in packet allocation is attainable. This provision considerably strengthens the result of equilibrium and provides the foundation for simulation and verification of congestion control in Model 2.

5.2 Contribution of PROMELA/SPIN in Model 2

With the addition of an administrator process, PROMELA/SPIN provide an important mean to simulate runtime dynamic price adjustment in the priority pricing scheme. Model 2 distinguishes itself from earlier work on dynamic price adjustment [4] in two important ways. Using the simulation-verification capability provided by PROMELA/SPIN and the results obtained from Model 1, each inter-price-change segment in Figure 4 can be considered as an independent simulation similar to Model 1. Therefore, the existence of an equilibrium for each segment and its convergence can be formally verified. In addition, data collected from the simulation of Model 2 also indicates an inverse proportionality between price and total packets allotted to network users. As illustrated in Figure 4, when price is high, the aggregated demand rests on a lower volume in equilibrium. This results obtained from PROMELA/SPIN is arguably stronger than similar results based on simulation alone. Further investigation and verification of this inverse price-volume relationship can be readily supported by PROMELA/SPIN. In this regard, congestion control studies based on priority pricing modelling using PROMELA/SPIN clearly presents a unique advantage over other simulation tools.

5.3 Limitations

In Model II we have performed verifications using only one set of initial conditions, namely all user processes begin with a zero allocation to all priority classes. Given another set of initial allocations, it would be desirable to observe an identical equilibrium condition. The ultimate goal is to verify the more general case where proposition 1 and 2 remain affirmative for all sets of initial conditions. Little evidence is observed where PROMELA/SPIN would be limited in verifying the general case. However, it may not be feasible to formulate each user process in LTL. In spite of this, the simulation and verification framework provided by PROMELA/SPIN is still stronger than using simulation alone with the currently available network tools.

We have compared the applicability between SPIN versus SMV in modeling priority pricing. In SMV, it is possible to model all users updating packets allocation in one single time slot. When modeling in SPIN, we are forced to simulate the interaction between user and the network asynchronously. In each time slot, only one user process operates. In this regard, SMV's capability to model synchronous system seems to make it a more logical choice. After experimenting with both automated verification tools, we have chosen PROMELA/SPIN for PROMELA's expressiveness and SPIN's verification capability. We are especially impressed with the rich language constructs and programming flexibility provided by PROMELA.

6 Conclusion

Throughout this work, we have been impressed with the expressiveness of PROMELA and interpretative strength of SPIN as a simulation-verification tool. We have been equally impressed with the strength of automated verification tools in general in providing modeling and verifying capability to complicated scenarios. By following the assumptions made [1], we successfully developed three process types in SPIN, interacting with one another asynchronously in simulation of a network link in action. Not only have we verified in SPIN that the postulated equilibrium indeed existed, we have shown in Model 1 that convergence to equilibrium can be achieved. Furthermore, we demonstrated in Model 2 the unique result that effective congestion control can be achieved through dynamic price adjustment. The choice of using PROMELA/SPIN for this work had arisen through necessity. The strength of combined simulation and verification framework provided by PROMELA/SPIN would be difficult to replicate using another tool.

Acknowledgement

We would like to express our gratitude to Professor Marsha Chechik for her invaluable suggestions and patient guidance throughout this work. We are especially indebted to her tireless effort that had led us through numerous revisions of this manuscript. We would like to thank Professor Peter Marbach for his guidance in the understanding of the priority pricing model for congestion control.

References

[1] P. Marbach, "*Pricing Priority Classes in Differentiated Services Networks,*" 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 1999.

[2] A. Gupta, D. O. Stahl, and A. B. Whinston, "*A Priority Pricing Approach to Manage Multi-Service Class Networks in Real-Time,*" Presented at MIT Workshop on Internet Economics, March 1995.

[3] A. Gupta, D. O. Stahl, and A. B. Whinston, "*The Economics of Network Management,*" Communications of the ACM, 42(9):57-63, September 1999.

[4] A. Gupta, D. O. Stahl, and A. B. Whinston, "*A Stochastic Equilibrium Model of Internet Pricing,*" Journal of Economics Dynamics and Control, 21:697-722, 1997.

[5] G.J. Holzmann, "*The Model Checker SPIN,*" IEEE Transactions on Software Engineering, 23(5):1-17, May 1997.

[6] E. Fersman and B. Jonsson, "*Abstraction of Communication Channels in PROMELA: A Case Study,*" In SPIN Model Checking and Software Verification: Proc. 7th Int. SPIN Workshop, volume 1885 of Lecture Notes in Computer Science, pages 187-204, Stanford, CA, 2000. Springer Verlag.

[7] Theo C. Ruys and Rom Langerak, "*Validation of Bosch' Mobile Communication Network Architecture with SPIN,*" In Proceedings of SPIN97, the Third International Workshop on SPIN, University of Twente, Enschede, The Netherlands, April 1997.

[8] Klaus Havelund, Thomas Pressburger, "*Model Checking JAVA Programs using JAVA PathFinder,*" STTT 2(4):366-381 (2000)

[9] Wil Janssen, Radu Mateescu, Sjouke Mauw, Peter Fennema, Petra van der Stappen, "*Model Checking for Managers,*" SPIN 1999: 92-107

[10] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu, "*Advances in Network Simulation,*" IEEE Computer, 33 (5), pp. 59-67, May, 2000.

[11] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala, "*Improving Simulation for Network Research,*" Technical Report 99-702, University of Southern California, March, 1999.

[12] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, and Haobo Yu, "*Network Visualization with the VINT Network Animator Nam,*" Technical Report 99-703, University of Southern California, March, 1999.