A Rate Allocation Protocol Using Competitive Pricing for Improving Performance of Multicast Sessions

Zohar Levy and Danny Dolev

Institute of Computer Science, The Hebrew University of Jerusalem,

Givat Ram, Jerusalem 91904, Israel

ABSTRACT

Rate allocation using the Max-Min fairness criterion may highly discriminate against multicast and long unicast sessions and may lead to severe network underutilization. In this paper, we present a solution for rate allocation that is based on competitive pricing. The resultant allocation increases fairness towards multicast sessions and improves network utilization considerably. The solution requires no re-routing of sessions. The economy on which we base our solution is simple enough, enabling its implementation for practical use. We present a distributed asynchronous protocol suitable for the ATM ABR service, which achieves the economy's allocation efficiently and with short convergence time.

Keywords: Rate Allocation, ABR, multicast, dynamic pricing

1. INTRODUCTION AND MOTIVATION

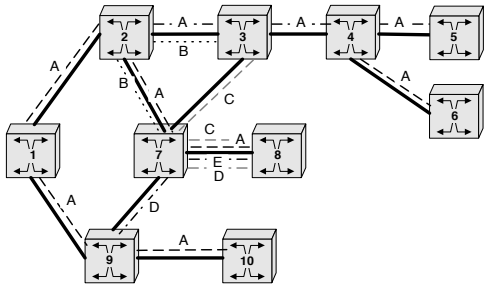
Dynamic network resource allocation is a basic building block for supplying efficient network services to multiple users. In the past several years, numerous distributed protocols for computing fair and efficient network resource allocation have been devised, along with the development of the ATM networking technology. A vast majority of the effort has concentrated on rate allocation schemes that accommodate point-to-point (unicast) sessions. Several papers present technical modifications to existing allocation schemes that enable the support of point-to-multipoint (multicast) sessions.[1,2] However, the question of overall fairness and network utilization in the presence of such sessions has hardly been tackled.

One of the more commonly accepted fairness criteria that is used today for rate allocation is the *max-min fairness* criterion.[3,4] Max-min fairness attempts to allocate equal rates to all active sessions while keeping the allocation *feasible* - the total bandwidth allocated by a link is always equal or lower than its capacity. According to max-min fairness, a feasible rate allocation is fair if the rate r_s allocated to a session s cannot be increased without decreasing the allocation $r_{s'}$ of a session s' where $r_{s'} < r_s$. In other words, the minimal rate allocated to a session by max-min fairness is no less than the minimal rate allocated by any other feasible allocation. The second smallest rate allocated by max-min fairness is no less than the second smallest rate allocated by any other feasible allocation and likewise for the third smallest rate and so on. This property of max-min fairness also imposes *Pareto-optimality*: once rates are allocated to the sessions according to this criterion, the rate of any session cannot be further increased without decreasing the rate of another session.

The introduction of multicast sessions poses many new questions concerning the fairness criterion and the implementation of the rate allocation scheme. Destinations (or receivers) in a multicast session may be located at different parts of the network, where there may exist different congestion levels. Could the source transmit at different rates to different receivers? If not, at what rate should it transmit - i.e., what rate would achieve both fairness and efficiency without causing excessive loss of data?

In order to avoid complex hardware solutions,[5] we confine ourselves to solutions of one rate per session, as in the point-to-point session model. Does the max-min fairness criterion provide a good compromise between fairness and network utilization when it is applied to multicast sessions, as well as to long point-to-point sessions? Applying max-min fairness for allocation in the presence of multicast sessions has two main drawbacks:

Other author information:
Z.L. (correspondence) :E-mail: Zohar.Levy@intel.com or zohar@cs.huji.ac.il Telephone: ++972-2-5892418
D.D.: E-mail: dolev@cs.huji.ac.il Telephone: ++972-2-6584116



	Maxmin Rate
Session A	1/4
Session B	3/4
Session C	1/4
Session D	1/4
Session E	1/4
Total Utilization	45.45%

All link capacities are 1

Figure 1. Bad Network Utilization Resulting From Max-Min Fairness Allocation

1. Max-min fairness *discriminates* against sessions that traverse many links, such as long unicast sessions and multicast sessions. This is because the more links a session needs to traverse, the higher the *probability* that it will be bottlenecked at a highly congested link and be assigned a low transmission rate. Max-min fairness does not take the number of links of a session into account.

2. The max-min fairness allocation may lead to *network under-utilization*: if a multi-hop or multicast session is bottlenecked in only a small number of links, and is limited it to a low transmission rate, most of the session's other links might be left unutilized.

An illustration of the severe consequences that may result from applying max-min fairness in the presence of multicast sessions can be seen in the following example, depicted in Figure 1. In order to be able to quantify utilization, we define a link's *usage* as the sum of the rates of all the sessions passing through it. We define the *total network utilization* as the sum of all link usages divided by the sum of the link capacities of all links that have a positive usage (i.e., unused links are not counted). In this example there are five active sessions and all the link capacities equal to 1. Four of these sessions (B,C,D,E) are relatively short passing 1-2 links each. Session A is a large multicast session traversing 9 links. It can be seen that sessions A,C,D and E are bottlenecked at the link between the switches $SW7$ and $SW8$. Thus, according to max-min fairness, each of these sessions must receive an equal portion of the link's capacity, i.e., $\frac{1}{4}$. Session B benefits from session A's low bottleneck value and receives an allocation of $\frac{3}{4}$. The total measured network utilization resulting from this allocation is 45.45%. Any increase of Session A's allocation (on the expense of the rest of the sessions) will improve the total network utilization because Session A passes through many links in which it is the only bandwidth consumer. At the extreme case we could allocate all the available bandwidth to Session A to reach 100% network utilization. This solution, of course, is unfair to the rest of the sessions.

The fact that the max-min fairness allocation tends to allocate low transmission rates to multicast sessions may produce two additional undesired side effects. Firstly, data-oriented multicast and long unicast sessions (such as those that use the ATM ABR service) will require more time in order to transmit their data. Thus, although such sessions transmit at lower bandwidth during their lifetime, they hold up resources for longer periods of time. Secondly, as more leaves are added to a multicast session, its rate decreases. In some scenarios, the rate of a multicast session can radically drop as a result of the addition of just a single leaf to the session. This property of the max-min fairness allocation could be quite problematic for servers that use multicast connections for serving large groups of clients and desire to maintain a stable service level.

This discussion suggests that in order to improve fairness towards multicast session and boost overall network performance, the rate allocation given to multicast sessions should be increased. However, as hinted in the above example, this must be done with care in order not to shift unfairness to the other direction. In addition, as will be demonstrated later in this paper (in Figure 2(b)), there are scenarios in which increasing the allocation of long sessions may *decrease* overall network utilization. Thus, a solution that takes this approach must *avoid* increasing the rate of a session if this decreases network utilization, or if it causes a degradation in the performance of too many other sessions. Such a solution is described in the next section.

2. SOLUTION MODEL: THE NETWORK BANDWIDTH ECONOMY

The model we use for our solution, *The Network Bandwidth Economy* (or simply, *the bandwidth economy*) views the network as a *very limited* competitive economy in which links are suppliers of resources and sessions are consumers. The model is based on the *network economy* model presented in.[6] However, our model is simpler in that sessions are strictly *throughput oriented*, i.e., they do not attempt to optimize other goals (e.g., minimize delay). This greatly simplifies the economic process and makes it possible to actually implement a protocol which calculates the economy's allocation, and is suitable for high speed networks (as will be seen in Section 4).

2.1. The Economic Process

Each session is modeled as having a predefined path, and an *endowment* of income. The sessions use their endowment in order to "purchase" resources from the links on their path and maximize their throughput goal. Each link offers a fixed *supply* of bandwidth and advertises a price for the usage of its buffer space. Sessions present their *demand* for resources in accordance with these prices.

The way a session s calculates its demand, given the advertised prices of the links it traverses, is straightforward. The sum of these prices, denoted by Q_s, represents the amount that session s has to pay for each unit of bandwidth it consumes on its path. Since s's rate is limited to the minimal rate allocated to it by a link on its path, s will demand the same bandwidth from each link on its path. Thus, given Q_s and an endowment of W_s, s can afford to demand a rate allocation of $\frac{W_s}{Q_s}$ from each link (making the total payment sum up to exactly W_s). A full formalization of the demand calculation in the bandwidth economy can be found in .[7]

The links adjust their prices in response to the sessions' demands. If a link's resources are not fully utilized, i.e., it experiences *excess supply*, the price is lowered. If there is more demand for the link's resources than it can supply, i.e., there is a situation of *excess demand*, the price is raised. The sessions in response again calculate their demand according to the new prices, and so on. This process continues until the economy converges to a state of *equilibrium*.

In a general economy, the state of equilibrium is defined as the state in which all suppliers measure an excess demand/supply of zero. In the bandwidth economy this would mean that at each link l all the capacity is demanded by the sessions passing through l and the demands of all sessions can be satisfied. However, there are cases where no matter how low prices fall, the demand for the resources of a link l will remain lower than its capacity, because all the sessions passing through l are bottlenecked or physically limited by other links in their path. For this reason, equilibrium in the bandwidth economy is only achieved when for each link l either: 1) Excess supply/demand in l equals to zero, or 2) the price p_l presented by l falls below a predefined minimum ε (this does not limit generality since ε can be set to a value as small as desired).

By relying on economic theory, it can be shown that for any given topology and session configuration, there exists a set of prices that yields an *equilibrium* state.[6] There are cases in which an infinite number of equilibrium price vectors exist for a given topology and session group, but it is not determined whether more than one equilibrium allocation is possible. It should be noted however, that no example for multiple equilibrium allocations has been found in simulations made either in,[6] or in the framework of this work.

2.2. Endowment Allocation, Fairness, and Network Utilization

Using the economy's allocation process, we would like to find an endowment allocation that will produce a fair and efficient rate allocation. Before we can say whether a rate allocation is "fair" or not, we must define the *utility* of a session, i.e., what makes a session 'better off'. The utility of a throughput-driven session can be estimated either a) as the final transmission rate assigned to the session, or b) as the total amount of buffer space allocated to a session on the switches that the session traverses. The final transmission rate is more important from the user's point of view. However, from the "social" point of view, total buffer allocation is also significant, in that buffer space used by one session cannot be used by other sessions. It is generally assumed that this conflict will be resolved in public networks through some billing mechanism. Thus, as in the approach of max-min fairness rate allocation, we concentrate on achieving a fair allocation of final transmission rates and leave the matter of market resource pricing to mechanisms that are external to our model.

The endowment allocation we shall investigate in the paper is the *equi-length endowment* allocation: Each session receives **1 unit** of endowment *for each link* it traverses. Fairness is achieved in that each session receives an endowment that is proportional to the expected price it must pay for each unit of bandwidth it will utilize. This is under the

generalizing assumption that prices are randomly distributed and that a single session has a negligible effect on the price distribution. Two phenomena are expected to occur as a result of applying the equi-length endowment allocation: 1) Long sessions should improve their allocation in comparison to max-min fairness, and 2) Short sessions will tend to receive a higher allocation than longer sessions, but the difference is smaller than in the max-min fairness allocation. Unlike the max-min fairness allocation, in the economy's allocation longer sessions gain an advantage from passing through uncongested links. In uncongested links, the demand for the link's resources is low, so the economic mechanisms will drive down the price charged by this link for each unit of bandwidth. Thus, a session passing through many such links will have more of its income to spend on congested links, giving it in effect a higher priority in the congested links. As a result, sessions that are only congested in a small number of links will suffer less than in the max-min fairness model. However, the expected allocation for shorter sessions is still higher: If we view the links' prices as independent random variables*, the expected allocation given to a session passing k links is $E(\frac{k}{\sum_{i=1}^{k} p_i})$ (where k is also the session's endowment, and $p_1, p_2, ...p_k$ are the prices on the session's path) and is a decreasing function of k. A full proof of this claim is found in.[7]

It is important not to confuse the endowment allocation and pricing used by this economy with any external billing system imposed on the public network users. In the bandwidth economy, pricing is used only as an *internal* tool for computing an efficient and fair rate allocation. This view is identical to the view taken by the max-min fairness allocation model, which does not deal with external billing mechanisms or payments that will be charged for the use of network resources.

It should be observed that in the above endowment allocation scheme, or in any scheme that gives priority to longer sessions, it may be possible for a session to improve its transmission rate 'unjustly' by making its path from the source to the destination longer than necessary or by adding receivers to its multicast tree. To avoid this, users must not be allowed to affect routing decisions, and must be discouraged from adding unnecessary receivers to a multicast tree (e.g., through a billing mechanism).

3. SYNCHRONOUS SIMULATION RESULTS

For the purpose of examining the properties of the Bandwidth Economy's allocation in different network topologies and session configurations, we have run a set of *synchronous* simulations. Our main focus in these simulations was to compare the fairness properties and network utilization resulting from the economy's allocation to those achieved by the max-min fairness allocation. In the simulation set, network delay is ignored. Demand calculations made by the sessions and price adjustments made by the links are performed in synchronous steps. This means that all sessions present their demand at the same instant, and all price updates by the links are calculated at the same time.

The price adjustment scheme used is the *tatonement* process. Each link adjusts the price according to the previous price, its own capacity, and the measured excess demand, as follows:

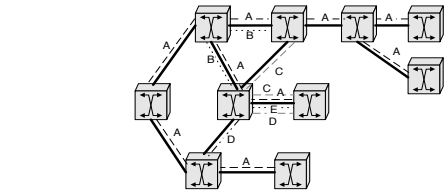
$$p = Max[p + \alpha \cdot p \cdot \frac{excess\ demand}{capacity}, \varepsilon],$$

where ε is the minimal allowed price and $0 < \alpha \leq 1$ is a predefined constant. This algorithm is shown to provide fast convergence to equilibrium and quick reaction to transient events†.

Two simple examples of the economy's behavior are shown on Figure 2. On Figure 2(a), it can be observed that the economy improves the allocation of the multicast session and overall network utilization in comparison to max-min fairness (this configuration was previously seen in Figure 1). Figure 2(b) demonstrates that the economy does not favor longer sessions blindly. In this case, Session A is the longest session, but the economy decreases its allocation because it traverses only congested 'expensive' links. In this case as well, the overall network utilization was improved.

*This assumption approaches reality as the network gets larger, when a single session has negligible effect on overall pricing, and session configurations are random.

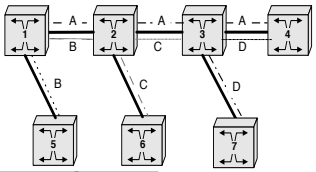
†The tatonement process is also used in.[6] Although it was not analytically proven that this scheme always leads to convergence of the economy, no example of non-convergence has been found for the economy described there or for the bandwidth economy. This is left for further study.



	Max-Min Fairness	Economy
Rate A	1/4	45/80
Rate B	3/4	35/80
Rate C	1/4	14/80
Rate D	1/4	14/80
Rate E	1/4	7/80
Total Utilization	45.45%	61.13%

All link capacities are 1

(a)



	Maxmin Rate	Economy Rate
Session A	1/2	1/3
Session B	1/2	2/3
Session C	1/2	2/3
Session D	1/2	2/3
Total Utilization	75.00%	83.33%

All link capacities are 1

(b)

Figure 2. Comparison between the network utilization achieved by the Max-Min fairness and the Economy's allocations when congestion is non-uniform. In Figure (a), Session A is a multicast session while the others are short unicast sessions. Session A is congested on a single link and therefore the economy improves its allocation. In Figure (b), Session A is the longest session but receives less in the economy's allocation.

In order to achieve a statistical insight of the economy's effects on rate allocation, we executed a large set of random configurations. In all of our simulations, the network had a fully connected topology, and session paths were chosen at random. This allowed testing of the economy in the most general cases (full results of these simulations is found in [7]).

Several observation were acquired from this set of simulations. The first of these was that, in comparison to the max-min fairness allocation, the bandwidth economy's allocation was shown to *reduce the unfairness* towards long and multicast sessions. Although short sessions received higher average rate allocations than long sessions in both allocation schemes, the difference was considerably smaller in the economy's allocation. For example, in a network of 220 links (all having a capacity of 1 unit) we randomly generated configurations with an average of 50 sessions, each passing an average of 10 links. Sessions passing 20 links received 15% more from the economy than by max-min fairness, while sessions passing 2 links received 12% less from the economy than from max-min fairness. The improved fairness was observed most clearly in cases of high congestion variation, as well as in configurations in which there were more active short sessions in comparison to long sessions.

The second significant result was that in a large majority of the cases, the network economy improved overall network utilization in comparison to max-min fairness. In uncongested networks there were cases in which the improvement of the overall utilization by the economy was extremely high, reaching 50%-60%. As the network was set to be more congested, the ability of the economy to increase utilization was reduced. High variations in link capacities also reduced the effectiveness of the economy. In the average case (again, in a network of 220 links of unit capacity), the economy improved utilization by about 7%. This figure becomes even more impressive when considering the fact that it was achieved without adding physical resources to the network and without re-routing of sessions. The cases in which the economy *decreased* network utilization were very rare (1%-2%) and the negative effect was negligible (again, 1%-2%).

4. ASYNCHRONOUS PROTOCOL IMPLEMENTATION

The algorithm presented in Section 3 for price update calculation assumes that all demand requests are presented by the session sources simultaneously, and that all the price updates of the links are also performed at the same time. Moreover, at each link the excess demand measured just before the next price update fully depicts the effect of the previous price vector on the demand of the sessions. These assumptions obviously do not hold in a real network environment. Network delay and asynchrony might delay the propagation of price/demand information, which in turn might lead the economy to high price oscillations, congestion and periods of network under-utilization.

In order for the bandwidth economy to be suitable for practical use, it must be able to operate in an asynchronous network environment. In this section we present a distributed asynchronous protocol for the economy. The protocol requires no per-VC queueing, and only minimal per-VC accounting. As in the ABR standard specification,[8] Resource

Management (RM) cells are sent by the source every fixed number of data cells, and are reflected back to the source by the destination. These cells carry price and demand information (instead of explicit rate information and congestion indication bits, as in the ABR standard). Each RM cell is forwarded transparently by the switches on its forward trip, and accumulates link prices on its backward trip. When the RM cell reaches the source, it contains the *sum* of the prices of all the links on the session's path. The RM cell format can be seen in Table 1. When multicast sessions are active, special actions are taken by the protocol in order to ensure that a single price reaches the source, and that the volume of backward RM cells does not depend on the number of branches.

In the rest of this section, we give a more detailed description of the protocol. The full protocol, including pseudo-code, can be found in.[7]

Table 1. RM cell format. Only fields relevant to exchange of economic information are displayed.

DIR	Direction of cell (FORWARD or BACKWARD)
DEMAND	Demand in rate units (e.g., cells per second)
PRICE	Sum of prices set by links on path
NUMLINKS	Number of links on path
STAMP	Binary value ‡

4.1. The Basic Protocol

For the implementation of the basic (point-to-point) protocol, each session maintains both its *current demand* and its *current rate*. For each of its outgoing links, each switch maintains a *price* and the current *excess demand*. The excess demand of a link is calculated as the difference between the link's capacity and the sum of the demands that were most recently recorded by the switch from the RM cells of all of the sessions passing through that link[§]. In addition, the switch maintains the value of each session's most recent demand and a *stamp* bit (whose function will be described below). This is the only per-session information maintained by the switch.

At initialization, the links set initial prices and each session has an initial demand as well as an initial rate (these are typically equal at this phase). After every *Nrm* data cells (typically 32), a session transmits a *forward* RM cell with its current demand, towards the destination. The cell also carries a price field initialized to zero. The cell is forwarded transparently by the switches towards the destination. Upon reaching the destination, the cell is reflected and sent back towards the source as a *backward* RM cell. On its return trip, the RM cell is examined by each switch on the path. The excess demand of the relevant link is updated and then the price of that link is updated according to the tatonement process formula (as described in Section 3).

In order to overcome the effects of network delay and asynchrony, several measures must be taken by the switch when calculating the price:

1. Instead of using only α in the tatonement formula, we use two constants α_{pos} and α_{neg} where $\alpha_{pos} > \alpha_{neg}$. The former is used when excess demand is positive and the latter is used when excess demand is negative. This is done in order to reduce congestion caused by too fast a price decrease, while allowing for fast price increase when necessary.

2. The constants α_{pos} and α_{neg} must also depend on the number of sessions passing through the link. This is necessary in order to mitigate price over-adjustment caused by the fact that each session presents its demand at a different time (thereby causing an updating step each time). Thus, both of these values must be reduced as the number of the sessions passing through the link gets larger[¶].

[§]A switch sw counts a session as passing through one of its outgoing links l only if sw is the switch feeding l with the session's cells in the downstream direction. This ensures that for each session only one switch maintains the price for each link the session passes.

[¶]A simple way to achieve this is to divide α by the number of active sessions. In the simulations shown in this paper, a more precise and slightly more complex formulation[7] was used.

3. When a new session becomes active, its initial demand is naturally calculated without consideration of the current prices. The switch must ensure that the initial price that a session sees when activated has a value that is not too small, in order to avoid too high a demand of the new session in reaction to this price. Although this is not required in order to guarantee the convergence of the protocol, it is needed in order to reduce the time of convergence. We have used in our simulations the value of $\frac{k}{c}$, where k is the number of active sessions passing thought the link, and c is the link's capacity.

4. In order to avoid infinite queue buildup, the length of the queue must also be considered when calculating the price. The longer the queue is, the higher the price should be. To achieve this, the switch multiplies the instantaneous queue length by a factor Q_FACTOR, and adds it to the calculated excess demand when calculating the price.

After the switch updates the price of the relevant outgoing link, the new price is added to the price already set on the RM cell and the new sum is then placed on the RM cell. The RM cell is then forwarded to the next upstream switch in the session's path. When the cell reaches the source, the source calculates its demand as $\frac{numlinks}{price}$ where *price* is obtained from the incoming RM cell and *numlinks* is the number of links on the session's path.

Once the demand has been calculated, the session's rate is adjusted accordingly: If the demand is lower than the current transmission rate, the rate is reduced to the value of the demand. If the demand is higher than the transmission rate then the rate is allowed to be increased gradually (at most by a factor of 2), unless the demand's measured standard deviation is low enough (i.e., it approaches stability). In such a case the rate is allowed to catch up with the demand.

Note that as opposed to the various actions taken by the switches to adjust prices in the asynchronous environment, the demand calculation made by the sessions is straightforward and is the same as in the synchronous algorithm. In other words, we have not taken steps to reduce demand oscillations directly through the demand calculation at the session's source. The reason for this is that the demand provides important information to the switches about 'the state of the market' and how prices should be adjusted. If we let considerations other than prices affect the demand, we might slow down convergence, and in some cases, never reach a Pareto-optimal equilibrium.

One problem that arises due to network delay, is that a switch might receive from a session several backward RM cells, which hold the session's demand relating to the same price. More specifically, a session could send an RM cell stamped with a demand D resulting from a price p, and before receiving knowledge of the price update resultant from this demand, the session again could send an RM cell with the same demand. This would cause over-adjustment of prices. For example, if, as a result of a session's demand, a link suffers from excess demand and raises its price, then receiving the same demand from that session would cause a further rise in the price, before the session has a chance to react to the first price raise.

In order to avoid this, the frequency of price updates made by the switches must be limited. We define *RM-RTT* (RM round-trip time) as the time needed by the source to learn about a price update performed in a switch, plus the time needed for the switch to learn about the reaction of the source to that price update (i.e., a demand change). A switch may update its price according to a session's incoming RM cell *at most* every RM- RTT (which approximates to the RTT of that session within a margin of Nrm cell times). Each source maintains a *stamp*, which is a binary value that is placed on a bit on outgoing RM cells. This bit is inverted every time an RM cell carrying this stamp value has completed a round-trip (meaning that at least one 'normal' RTT elapses between consecutive stamp-inversions). A switch may only update the price of a link as a result of an incoming RM cell if the stamp on that RM cell is different from the most recent stamp previously received from that session (i.e., an RM-RTT has elapsed). However, the switch keeps updating its measured excess demand according to every incoming RM cell. It is not guaranteed (or necessary to guarantee) that during the whole RM-RTT of a session the prices on its path will not change (as a result of other sessions' demand) or that the demand of the session will not change (as a result of these price changes).

The protocol we described is tolerant of cell loss: Loss of some RM cells does not alter the correctness of the outcome rate allocation. In the worse case, a source will keep transmitting RM cells with the same stamp value, until one of these cells completes a round trip. Since the stamp remains the same, the transmitted cells will not cause further price updates at the switches until the source has received knowledge of the previous price updates. The effect of a lost RM cell on a demand calculation at the source will be cancelled once another RM cell completes a round trip.

4.2. Protocol Enhancements for Handling Multicast Sessions

The protocol described so far is not sufficient for handling multicast sessions. If we leave the protocol unmodified, branching points of multicast sessions will simply multicast forward RM cells (just as is done for data cells) and will forward every backward RM cell received from a downstream branch back to the source. Two immediate problems result from this behavior. Firstly, backward RM cells arriving to the source from different branches will hold different prices. Secondly, the volume of the backwards RM cell traffic will depend on the number of receivers of the session. Apart from the additional overhead, the higher amount of backward RM cells received in intermediate switches will cause prices to be adjusted too frequently. A variation of the second problem exists for other rate-based algorithms such as the EPRCA[9,10] algorithm used for flow control in the ABR service of ATM. There too, the behavior of the session's source depends on the rate of the incoming backward RM cells, when adjusting the rate according to the one-bit feedback on the RM cell.

To solve these problems, we adapt the *cell consolidation* concept.[1] The consolidation scheme can be outlined as follows. Each switch sw that serves as a branching point for a multicast session s maintains aggregated information for all the downstream links residing on the subtree of s's multicast tree, whose root is sw. Whenever sw receives a forward RM cell for s it multicasts it transparently towards s's destinations. In addition, it immediately transmits a backward RM cell with the aggregated information toward s's source. Whenever a backward RM cell for s is received from one of s's downstream links, the aggregated information is updated according to the information on this cell, and the cell is discarded. Thus, for each forward RM cell s's source transmits, one backward RM cell reaches the source.

Two contradicting phenomena occur when using cell consolidation. On one hand, information regarding links closer to the source reaches the source faster. On the other hand, even if there is no network delay present, a multicast tree with N levels of branching (i.e., there is a path to a destination that includes N branching points) will require N transmissions of forward RM cells by the source before information from destinations residing in the deepest branching level will reach the source. Thus, it takes longer for a multicast session to react to congestion conditions in links that are distant from the source than it would take a point-to-point session of the same length to react to the same conditions. This delay may also be accompanied by *feedback noise*,[2] meaning that feedback from different branches would be delayed by different amounts of time because of varying RTT. The protocol we present here behaves reasonably well in networks of reasonable size but may require further enhancements in order to reduce the feedback noise on very large networks.

In the economy's protocol, the primary aggregated information that a branching point sw must maintain for each branching session s is the sum of the prices P of the links belonging to s's multicast subtree, rooted at sw. Each time a backward RM cell arrives from one of the branches, P is updated and the cell is discarded. Each time a forward RM cell for a session s arrives, the cell is multicasted downstream. In addition, the prices of the directly connected links may be updated (depending on the stamp on the RM cell) and the sum of these prices plus P is put on another copy of the RM cell. This additional copy is sent to the next upstream switch on the session's path. Any 'normal' (non-multicast) switch that receives the backward cell adds the relevant price to the price on the cell (as normal) and transmits it upstream. The RM cell will be discarded either when it reaches the next upstream branching point of the session, or once it reaches the source.

The difference in the flow of RM cells in multicast sessions requires a change in the stamp mechanism as well. Information from branching points closer to the source reaches the source faster (i.e., the RM-RTT for branching points nearer to the source is smaller). This suggests that the prices of links closer to the source can be allowed to be updated more frequently. In order to work according to this policy, each branching point sw of a session maintains a separate *downstream-stamp* for each directly connected downstream link l. This stamp is inverted only after a) at least one round-trip between sw and sw' (the next downstream branching point in the direction of l) has been completed, and b) *following this*, the RM-RTT for sw has elapsed. Assuming that the RM-RTT of sw is estimated correctly, it can be deduced that the time between two consecutive inversions of the *downstream-stamp* by sw estimates the RM-RTT for sw'[‖]. Thus, as the number of branching points separating a link from the source increases, the frequency of its price updates decreases.

[‖]Given the fact that the first downstream switch from the source can calculate its RM-RTT directly (as in the point-to-point protocol), we can use induction to prove that every switch can calculate its RM-RTT from the source using the downstream-stamp generated by its upstream neighbor.

4.3. Simulation Results

In this section we present results of network simulations which we have performed in order to test the economy's protocol performance in an asynchronous environment. The simulations show that the protocol behaves well and overcomes most of the problems resulting from network delay and asynchrony. In unicast-only configurations such as the Generic Fairness,[11] convergence speed was comparable to that of algorithms that attempt to achieve max-min fairness.[12] In multicast configurations, convergence was fast as well, although somewhat slower than in the unicast case (due to feedback noise and the problem of delay in branching switches).

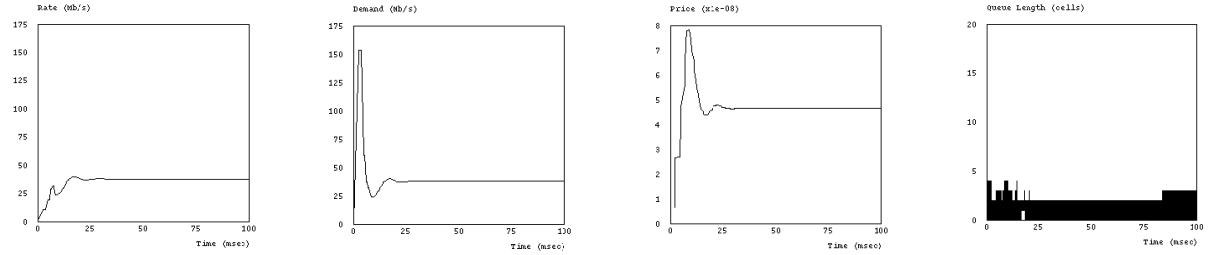
We have simulated the protocol using the OPNET ** network simulator. In all of our simulations, buffering in the switches is done in output ports. Cells are served in FIFO order. Cells suffer propagation delay only in transmission links (according to their physical length) and while pending in the buffer queues in the switches. Queue buffers at the switches are of infinite size and no cell loss is simulated.

Link capacities are measured in bits-per-second (bps). Links are full-duplex, so the marked capacity of a link is fully available in both directions. In order to enable faster queue clearance at times of congestion, only a maximum 95% of the link's capacity is allowed to be allocated to sessions. For the ease of presentation, the marked capacities represent the capacity available for allocation while the physical capacity is a bit higher (therefore, the marked capacity is 95% of the physical capacity) ††. We assume that the backward flow of RM cells does not cause flow-control problems. Thus, no prices are charged for links traversed on the backward path.

If not indicated otherwise, sessions are presumed to be greedy, i.e., they fully utilize the bandwidth allocated to them. The following constants have been set for the session sources, unless specified otherwise: Initial cell rate (ICR) = 1 Mb/s, minimum cell rate (MCR) = 10 kbps., peak cell rate (PCR) = ∞.

The following constants have been set for the switches (both multicast and non-multicast): $\varepsilon = \frac{1}{155200000}$ (155200000 bps is the maximum possible link capacity, ε is the minimum allowed price), $\alpha_{pos} = 1.0$, $\alpha_{neg} = 0.75$, Q_FACTOR = 100.

Due to space considerations, only representative results are shown. The first three simulations (Figures 3- 5) test the basic behavior of the protocol in a single-link configuration. In Figure 7 we test the protocol in a more general configuration - the Generic Fairness configuration. The last simulation, shown in Figure 8, shows the behavior of the protocol when multicast sessions are present. Further results may be found in.[7]

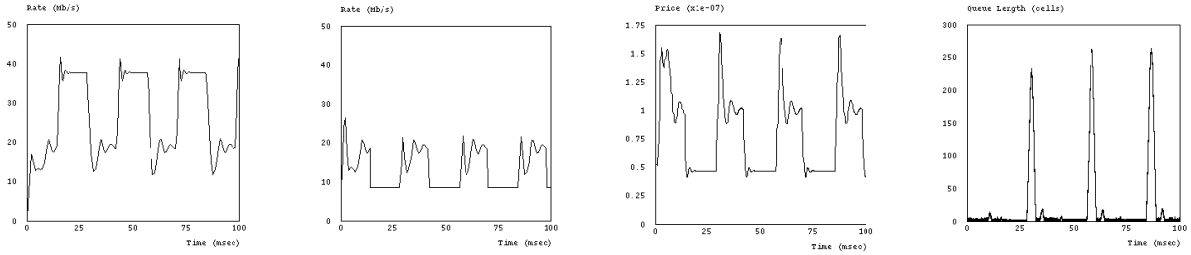


Rate of an arbitrary session. Demand. Price. Queue length.

Figure 3. The basic protocol behavior was tested on a single 151 Mb. link configuration. 4 sessions share the link. The RTT of each session is 2 msec. As in the max-min fairness allocation, each session deserves a fair share of the link - 37.75 Mb/s. The low initial demand of the sessions (1 Mb/s) sets low prices and, consequent high demand. This leads to momentary high transmission rates. The price rises again, and demands and rates drop, but to a higher point than the initial demands. The price and demand oscillations keep decreasing until equilibrium is reached. As can be seen from the right graph, queue buildup in this scenario is minimal. Convergence is achieved after about 20 msec. Sharp price and rate oscillations are avoided through the stamp policy, allowing the switch to update its price according to a session's demand only every RM-RTT of that session. Note that queue length remains small throughout the simulation.

**OPNET is a trademark of MIL 3

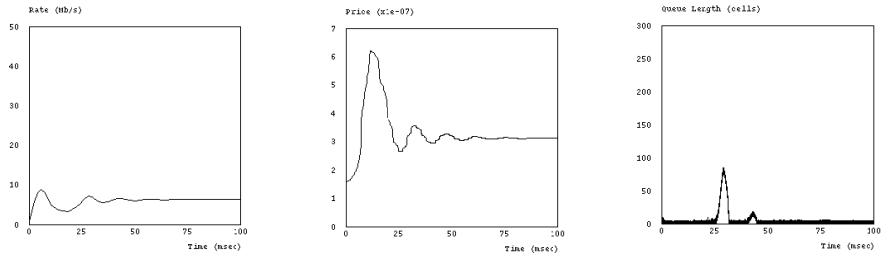
††For example, a link marked as having a capacity of 151 Mb/s, allows allocation of 151 Mb/s to sessions, but has an actual physical capacity of 159 Mb/s.



Rate of a greedy session. Rate of a bursty session. Price. Queue length.

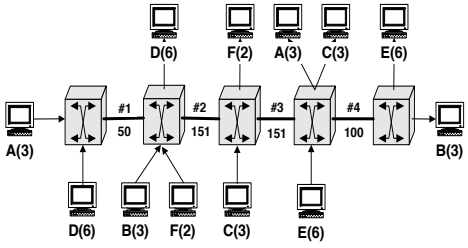
Figure 4. In this experiment, 4 greedy sessions and 4 bursty sessions share a 151 Mb link. The RTT of all sessions is negligible (0.01 ms). The bursty session transmits for 14 msec. and is idle for 14 msec. The ICR of the greedy sessions is 1Mb/s and the ICR of the bursty sessions is 8.5Mb/s. During the idle time of the bursty sessions, the 4 greedy sessions share the link among themselves, each transmitting at 37.75 Mb/s. During the active periods of the bursty sessions, each session deserves 18.9 Mb/s. At these periods the transmission rates are close to this value, although complete stability is not reached. Note that momentary queue buildup is formed every time the bursty sessions become active, but that no queue buildup is formed when these sessions become idle. Although the graph shows a rate of ICR for the bursty sessions during their idle periods, no actual data is transmitted by these sessions

.



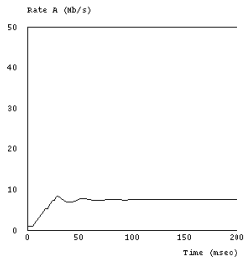
Rate of an arbitrary session. Price. Queue length.

Figure 5. In this experiment we examine the behavior of the algorithm when 24 sessions are concurrently activated at a 151 Mb link. The fair share of each session is equal to 6.292 Mb/s. The RTT of the sessions is negligible (0.01 ms). Initial queue buildup is avoided through the initial prices set by the link upon the entrance of each session. Upon the entrance of the k-th session, the price of the link guaranteed to be a least $\frac{k}{c}$ (where $c = 151,000,000$ is the capacity of the link in bps). A momentary (relatively small) queue buildup is formed after 25ms when the prices are slightly over-decreased, but recovery from this situation is fast.

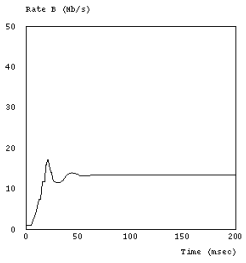


	Max-Min Fairness	Economy
Rate A	5.55 Mb/s	7.47 Mb/s
Rate B	11.11 Mb/s	13.33 Mb/s
Rate C	33.67 Mb/s	29.52 Mb/s
Rate D	5.55 Mb/s	4.59 Mb/s
Rate E	11.11 Mb/s	10.00 Mb/s
Rate F	50.50 Mb/s	44.28 Mb/s

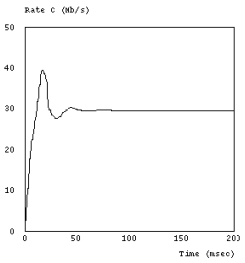
Figure 6. Generic Fairness Configuration. The letters indicate the ingress and egress nodes of each session group. The numbers in parentheses indicate the number of session in each group.



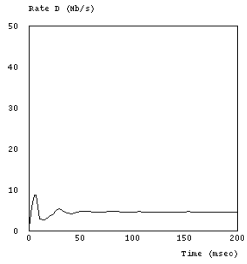
Rate of group A sessions.



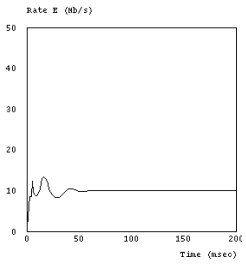
Rate of group B sessions.



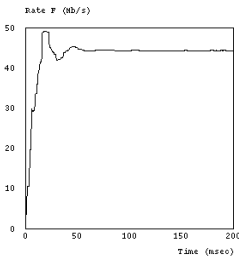
Rate of group C sessions.



Rate of group D sessions.

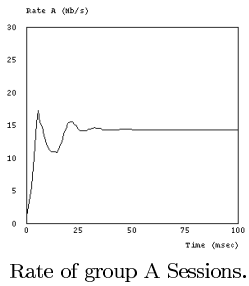


Rate of group E sessions.

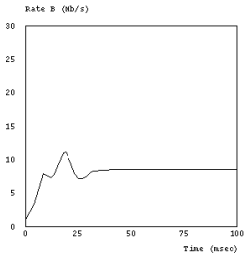


Rate of group F sessions.

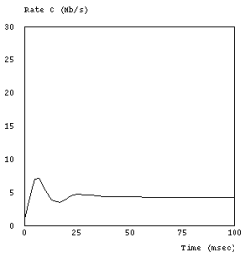
Figure 7. Here results of the protocol in the generic-fairness configuration[11] are shown. The configuration is depicted in Figure 6. In this simulation, the delay in the links connecting the switches is 1 ms, while the delay in the rest of the links is negligible (0.005 ms). Figure 6 also shows the expected rate allocation according to the economy in comparison to that of the max-min fairness scheme. Note that the longer sessions (A and B) receive a slightly higher allocation than in the max-min fairness allocation, on the expense of the rest of the sessions, which are shorter. This can be seen as a compensation to these longer sessions for the higher delay which they experience in their path.



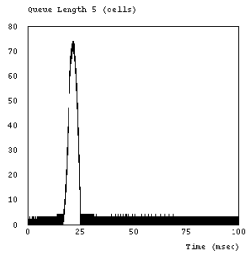
Rate of group A Sessions.



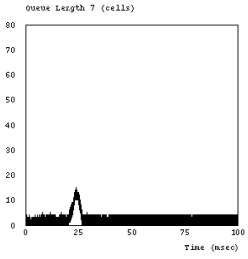
Rate of group B Sessions.



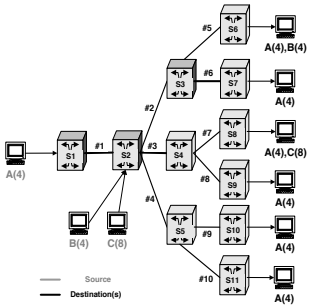
Rate of group C Sessions.



Link #5 queue length.



Link #7 queue length.



Multicast configuration.

Figure 8. In this simulation we test the behavior of the algorithm when multicast sessions are also present. In the topology shown on the right, sources are marked in a lighter shade. Four active multicast sessions suffer different congestion levels in different branches. In addition to the multicast sessions (marked 'A' in the figure), 4 additional unicast sessions marked 'B' pass through links #2 and #5, and 8 additional unicast sessions marked 'C' pass through links #3 and #7. The capacities of links #5-#10 are all set to 100 Mb/s. The capacities of the rest of the links are set to 151 Mb/s. The delay in all links is 0.005 msec. The economy allocates 14.9 Mb/s to each session in group 'A', 9.7 Mb/s to each session in group 'B', and 4.8 Mb/s to each session in group 'C'. All sessions approach their fair share after 25 msec.

5. CONCLUSIONS

We have presented a rate allocation scheme which reduces the discrimination towards multi-hop and multicast sessions that might result from the max-min fairness allocation. The algorithm computes the allocation using a competitive economy model, where sessions act as consumers and links act as suppliers of resources. Fairness is achieved by endowing each session with a budget that is proportional to the session's expected expenses. The economy's allocation typically leads to an increase in the overall network utilization in comparison to the utilization achieved by the max-min fairness allocation.

We have supplied a distributed asynchronous algorithm for calculating the economy's allocation. The algorithm overcomes demand oscillations by allowing session rates to gradually approach transmission rates until the economy approaches stability. The effects of asynchrony on convergence are handled at the switches through the maintenance of dynamic parameters.

There are several possible directions for future research: As stated earlier, although an equilibrium allocation exists for any network configuration, the convergence of the tatonement price-update algorithm has not been shown analytically to converge. Further investigation is necessary in order to answer this question in the framework of the bandwidth economy. In addition, the effects of other endowment allocation schemes on fairness could also be investigated. Finally, further work can be done on the asynchronous protocol in order to overcome the feedback noise problem and reduce memory requirements.

ACKNOWLEDGMENTS

I would like to thank Zvi Ostfeld and Prof. Yehuda Afek from the Tel-Aviv University for their helpful comments on the asynchronous protocol's implementation. Prof. Ariel Orda from the Technion in Haifa and Prof. Shlomo Yizhaki from the Department of Economy in the Hebrew University provided helpful guidance concerning the economic model. This work was partially funded by the Israeli Ministry of Science, grant number 032-7658, and by the Leibniz Center for Research in Computer Science.

REFERENCES

1. L. Roberts, *Rate Based Algorithm for Point to Multipoint ABR Service.* The ATM Forum Technical Comittee, Traffic Management Sub-working Group, September 1994. An ATM Forum Contribution 94-772R1.
2. S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, B. Vandalore, and X. Cai, *Performance Analysis of ABR point-to-multipoint connections for bursty and non-bursty traffic with and without background.* The ATM Forum Technical Comittee, Traffic Management Sub-working Group, April 1997. An ATM Forum Contribution 97-0422.
3. D. Bertsekas and R. Galager, *Data Networks*, Prentice Hall, Englwood Cliffs, N.J., 1992.
4. A. Charny, "An algorithm for rate allocation in a packet-switching network with feedback," Master's thesis, Massachusettes Institute of Technology, May 1994.
5. M. Tufail and B. Cousin, "Timer imposed and priority supported (TIPS) congestion control scheme for point-to-multipoint connections in ATM," December 1996. Onziéme congrés "De nouvelles Architectures pour les Communications, Large Bande: Internet ou RNIS". Versailles, France, December 1996.
6. D. Ferguson, C. Nikolau, and Y. Yemini, "Microeconomic algorithms for flow control in virtual circuit networks," in *Proceedings of the IEEE Infocom*, June 1990.
7. Z. Levy, "A rate allocation protocol using competitive pricing for improving performance of multicast sessions," Master's thesis, The Hebrew University, Jerusalem, Israel, September 1997. A postscript version can be downloaded form http://www.cs.huji.ac.il/labs/transis/thesis.html#thesis.zohar.
8. The ATM Forum Technical Group Members (Traffic Management)., *Traffic Management Specification Version 4.0 af-tm-0056.000*, April 1996.
9. A. W. Barnhart et al, *Closed-Loop Rate-Based Traffic Management.* The ATM Forum, September 1994. An ATM Forum Contribution 94-438R2.
10. L. Roberts, *Enhanced PRCA (Proportional Rate-Control Algorithm).* The ATM Forum, August 1994. An ATM Forum Contribution 94-0735R1.
11. L. Wojnaroski, *Baseline Text for Traffic Management Sub-Working Group.* The ATM Forum Technical Committee Traffic Management Sub-Working Group., October 1994.
12. Y. Afek, Y. Mansour, and Z. Ostfeld, "Phantom: A simple and effective flow control scheme," in *Proceedings of SIGCOMM*, pp. 169–182, August 1996.