

# Managing Bandwidth in Multimedia Applications using a Market-based Middleware

Johan Kristiansson, Jeremiah Scholl, and Peter Parnes

Department of Computer Science & Electrical Engineering  
Luleå University of Technology, 971 87 Luleå, Sweden  
{Johan.Kristiansson, Jeremiah.Scholl, Peter.Parnes}@ltu.se

**Abstract.** <sup>1</sup> This paper presents an application-layer middleware that applies a microeconomic model to help multimedia applications utilize available bandwidth in a way that maximizes the user's net benefit. The key components are a bandwidth broker that puts the supply of available bandwidth on a virtual market residing inside the application, and utility functions for each media, which are used to calculate their relative gain to the user at each bandwidth level. Basic supply and demand principles are used where the broker raises a virtual price if the total demand from all media exceeds the available supply, or lowers the price if demand is lower than the available supply. The advantage of the middleware is that it allows problems related to network management (usually affecting the supply) and human computer interaction (usually affecting demand) to be researched and integrated separately into an application and combined to leverage bandwidth in the best possible way. As a proof of concept, a prototype has been built by integrating the middleware into Marratech Pro, a commercially available e-meeting application. The paper presents experimental results using this prototype.

## 1 Introduction

Distributed multimedia applications provide users with a variety of inherently dynamic media, each having bandwidth requirements that can rapidly change over time. While a significant amount of research has targeted the creation of specific media that can adapt to bandwidth fluctuations (e.g. layered video coding), a still relatively unsolved problem is how to obtain bandwidth from various networks during a multimedia session, and then share the bandwidth efficiently between the different media inside an application in order to provide the user with the optimal aggregated experience.

Solving this problem requires a large amount of interdisciplinary knowledge. First of all, in order to obtain bandwidth in the best way designers must be able to deal with an increasingly complex network infrastructure. For example, applications must be able to handle IP mobility and QoS requirements and also consider financial aspects when switching between different wireless networks. Secondly, since user-perceived performance depends critically on the way bandwidth is shared between various media,

---

<sup>1</sup> This work was done within the VITAL project, which is supported by the Objective 1 Norra Norrland - EU structural fund programme for Norra Norrland. Support was also provided by the Centre for Distance-spanning Technology (CDT).

designers must also deal with human factors in order to calculate the relative worth of each media stream to the user. For example, it might be useful to allocate more bandwidth to “important” users in a multimedia conferencing session [3], [15], or to allocate less bandwidth to a video stream to make room for an audio stream.

This paper presents a middleware framework based on microeconomic principles of supply and demand to deal with bandwidth related issues in multimedia applications. The middleware consists of a virtual marketplace that functions as a management layer for deciding how to best obtain bandwidth and how to best consume bandwidth. The advantage of the middleware is that it allows the various solutions related to network management (usually affecting the supply) and the various solutions related to usability (usually affecting demand) to be researched and integrated separately into an application. Ultimately, this will allow various experts from fields such as human computer interaction and computer communications to combine their knowledge so that bandwidth can be obtained and divided between several media streams in the way that provides users with the most benefit.

The rest of the paper is organized as follows. Section 2 covers previous work done in the area. Section 3 gives a brief introduction to microeconomics in relation to the problem. Section 4 gives an overview the middleware, and in section 5 the middleware is evaluated using a prototype implemented into a commercially available e-meeting application called Marratech Pro. In section 6, a summary and conclusions are given followed by future work.

## 2 Related work

A considerable amount of research has been carried out to provide QoS support in distributed multimedia systems [9, 13, 14]. In [13] K. Nahrstedt et al. give an overview of existing middleware systems that have been proposed to support applications in heterogeneous and ubiquitous computing environments. To name just a few efforts, Agilos (Agile QoS) [9] is designed to serve as a coordinator to control the adaptation behavior of all concurrent applications in an end system so that the overall system performance is maximized. Similarly, Q-RAM [14] proposes a method to allocate resources between applications so that the system utility is maximized under the constraint that each application can meet its minimum QoS requirement. In contrast to the middleware proposed in this paper, these middlewares are not based on the concept of a virtual marketplace and generally focus on sharing resources between applications running on the same machine, or in the same network, rather than utilizing available bandwidth in the best possible way between several media within the same application.

A variety of papers have been published that use microeconomics as resource management method for bandwidth in conjunction with real economies [4, 11, 12]. However, the mechanisms described are generally dependent on support from nodes within the network and/or on variable rate pricing schemes for bandwidth. Both of these requirements have drawbacks in that dependency on router support can make systems much more difficult to deploy and because there is a strong evidence that users find dynamic pricing to be unacceptable [4].

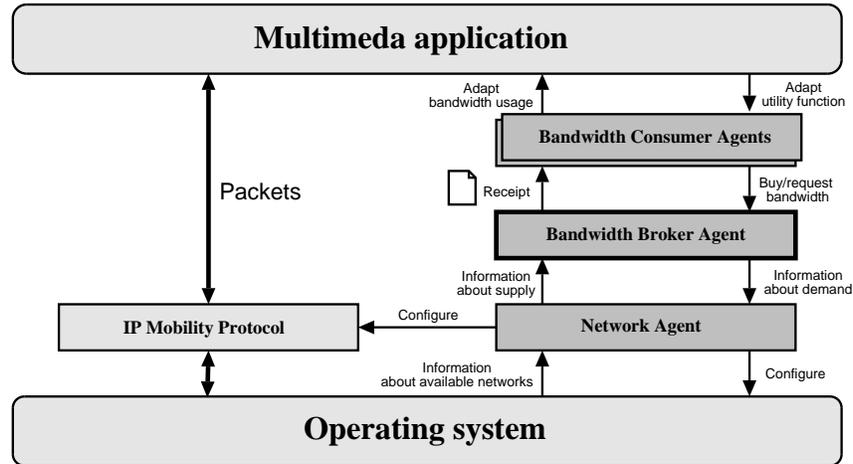


Fig. 1. Interaction between agents in the middleware.

The work presented in this paper differs from previous work in that the middleware uses microeconomic theory in a novel way by applying it inside multimedia applications without assuming the existence of non flat-rate pricing schemes for bandwidth or additional support from nodes within the network. Instead, microeconomics is used in order to run a *virtual economy inside the application* in order to make it easy to combine various network services, such as IP-mobility and congestion control while maintaining the efficient use of resources and maximizing the benefit to the user.

The idea of a market-based middleware was briefly mentioned in a poster paper [7] previously published by the authors. However, this paper contains a more extensive description of the middleware as well as a proof of concept implementation used in several experiments presented in the paper.

### 3 Market-based bandwidth management

Microeconomic theory deals with production, purchase and sales of commodities that are in limited supply [5]. In this context, the commodity on the market is bandwidth, and is traded by two key players: *consumers* and *suppliers*. Consumers attempt to optimize their gain by purchasing commodities (i.e. bandwidth) that give them the maximum gain at the lowest price, and suppliers try to sell commodities at the highest price they can get in order to maximize their own profit. This leads to a variable pricing system that works like an “invisible hand” in order to distribute and allocate resources efficiently despite the selfish actions of each player. Eventually, this price fluctuation will reach a state where the demand for goods at the going price equals the supply for goods at that

price. When this state is reached, all resources are fully utilized and the market is said to be in *equilibrium* [5].

In practice, equilibrium prices can be difficult to calculate because demand and supply vary over time. The supply will for example vary depending on the type of connection in use, congestion, financial constraints set by the user, or because of wireless interference. The demand may vary due to a wide variety of factors unique to every application. For example, in an e-meeting application the demand for the video stream of a particular user may vary depending on communication patters such as who is the current speaker [15].

One way of solving this problem is to use a *tâtonnement process* [5] to adjust the price iteratively until an equilibrium price is obtained. In this way, producers decrease the price if their production is not sold and increase the price if demand exceeds the supply.

$$p_{n+1} = p_n \cdot \frac{d}{s} \quad (1)$$

Equation (1) shows how the price is iteratively calculated based on supply and demand using the *tâtonnement process*, with  $p_n$  representing the current price,  $p_{n+1}$  next price,  $d$  the aggregate demand of all media, and  $s$  the current supply. As  $p_{n+1}$  is recalculated at discrete intervals of time, equation (1) will adjust the price towards a new equilibrium when either the demand or the supply changes. However, if the price is not recalculated fast enough, there is a risk that demand will not adjust to match the supply in time, which can either cause *over-demand* (over-utilization), or *under-demand* (under-utilization). In section 5 it is investigated how this affects the performance and the stability of the market.

## 4 Overview of the middleware

Middlewares are designed to manage complexity and heterogeneity in distributed systems and are defined as a layer above the operating system but below the application. Figure 1, shows an overview of the proposed middleware, which operates on the market principles previously described. The key player in the virtual marketplace is the *Bandwidth Broker Agent (BBA)*, which acts as a go between connecting all the buyers and sellers. Thus, the BBA sells bandwidth to all the different media streams used in the application, while obtaining bandwidth from various networks. Note that the middleware is implemented in the application-layer and *does not require support* from the network infrastructure or other clients/servers.

Each media has its own *Bandwidth Consumer Agent (BCA)* acting as its representative on the market for purchasing bandwidth. By using an optimization method described in subsection 4.2, the BCA calculates the total amount of bandwidth that should be purchased in order to maximize the benefit to the user. The BCA also communicates information to the media it represents regarding bandwidth it has purchased so that the media can adapt accordingly. For example, based on this information a video encoder will be able to change the video quality, or the interval at which it encodes frames.

The *Network Agent (NA)* contributes to the market by obtaining the actual supply of bandwidth that will be sold by the BBA. The purpose of the NA is not to actually

provide the bandwidth (e.g. requiring packets to be sent/received through the NA) but rather to make sure that the application is connected to the best available networks without explicitly requiring the user to manually configure the application or the operating system. Depending on services available to the application, the NA can be responsible for managing policy based routing, configuring mobility protocols, logging in to wireless networks, dealing with congestion control and so on.

In addition, the NA periodically receives information about current demand levels from the BBA, which can be used to make decisions on how to best obtain future supply of bandwidth. For example, if the current network round-trip-time is too high to be useful for a particular media, the BCA will reject sales offers from the BBA on the grounds that the product (bandwidth) is of too low quality. The BBA will then forward this information to the NA allowing it to take appropriate action (such as looking for a new network provider) if possible. Once the operating system has been correctly set up, the NA passes information about the available bandwidth to the BBA so that it can be sold to the various BCA. How the supply is calculated is described more closely below.

#### 4.1 Calculating the supply

The total supply that the BBA can put onto the market is directly related to the amount of bandwidth available to the application and can be bounded by a variety of factors. Most often the supply will be equal to the *bottleneck bandwidth* to the other end-point, but for non-flat-rate connections it may also be bounded by budget controls set by the user. For example, if the user specifies a maximum burn-rate in \$/s it would set an upper-bound on bandwidth supply.

K. Lai et al. [8] summarizes several techniques such as the *Pathchar* and the *Packet Pair algorithm* that can be used to accurately measure bandwidth. As the main usage of these algorithms is congestion control, it can be possible to let the NA obtain information about available bandwidth directly from the congestion control scheme instead of implementing an independent method for calculating the available network bandwidth.

However, due to the varying requirements for individual applications and media, there is no one-size-fits-all congestion control scheme that can be used by the NA. For multicast traffic the problem is a bit more complex than for unicast and has resulted in the creation of a wide variety of protocols. In general, these protocols follow one of two strategies, either relying on the sender to adapt its send rate in a way that serves the needs of the entire receiver set (sender-based congestion control), or relying on the sender to make many quality levels available concurrently through separate channels, allowing each receiver to "sign up" for the appropriate channel(s) independently (receiver-based congestion control). This means that for multicast traffic the type of bandwidth being supplied on the market will change somewhat depending on the underlying congestion control scheme in use. When using purely sender-based congestion control the BBA supplies bandwidth for the media that wish to send out packets on the network, whereas the BBA supplies bandwidth for packets to be received when a receiver-based congestion control scheme is in use. However, independently of which congestion control scheme that is being used by the application, the supplied bandwidth is always affected by the bottleneck bandwidth to the other end-point (or to the multicast group), assuming that the user has not set a budget control. Consequently, the BBA

will have to host two separate markets if the client is communicating with two different end-points or using both multicast and unicast.

An analysis of the exact congestion control scheme that should be used by each individual application in order to calculate bandwidth is outside the scope of this paper. Instead, the proposed middleware is intended to help solve the orthogonal problem of how to leverage the available bandwidth in the way that gives the user the most benefit.

## 4.2 Calculating the demand

In order for the BCA to decide how much bandwidth to buy given the price  $p$  per unit of bandwidth, it must calculate the relative gain the media can offer the user if allocated the amount of bandwidth  $x$ . This is done by creating utility functions for each media,  $m$ , where each utility function  $u_m(x)$  maps the gain with different bandwidth levels. Since each media wants to provide the user with the maximum net benefit (also known as the consumer surplus,  $CS$ ) at a given price level, it can calculate the amount of bandwidth  $x^{cs}$  to purchase by solving the problem,  $CS = \max[u_m(x^{cs}) - px^{cs}]$  as stated in [5]. The aggregate demand,  $d$ , is used to calculate the new price during each iteration in equation (1), and is calculated as the summation of the  $x^{cs}$  of each individual media.

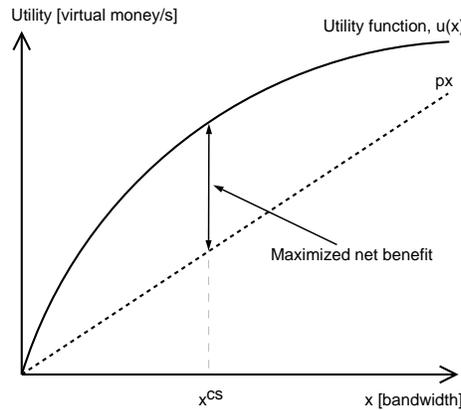


Fig. 2. A model for local optimization.

Figure 2 shows the relationship between a utility function,  $u(x)$ , and the total price,  $px$  it will cost the media  $m$  to obtain  $x$ . If the utility function is differentiable, strictly increasing and strictly concave for all  $m$ , C. Courcoubetis et al. [5] show that the maximize  $CS$  for media  $m$  can be found by calculating the  $x^{cs}$ , where

$$u'_m(x^{cs}) = p \quad (2)$$

Increasing concave utility functions are useful in this context since they give a fairly accurate model of media that are less sensitive to bandwidth changes when allocation reaches some maximum requirement [10]. Video is a good example of a media that falls

into this category since human beings can only notice a difference in the frame rate up until about 25 frames-per-second and tend to be more sensitive to changes below 15 frames per second. They tend to gain much more for example when raising the frame rate from 1 to 6 frames per second than from 20 to 25 frames per second.

Allowing utility curves to dynamically change based on contextual information available to the application is also possible. This allows for a high degree of customization to serve users more optimally under changing conditions. For example, for multimedia conferencing it has been proposed that video streams of certain “important” users should be prioritized by passing messages between clients in order to find out who is getting “attention” from the group [3, 15]. This type of scheme can be integrated into the market-based system by having each client use the information contained in these messages in order to change the utility curve for its video stream when appropriate.

Creating accurate utility curves for real world use may be a fair challenge, and therefore it is not expected that in most situations the user will be given this responsibility in any explicit way. However, application designers with a fair amount of expertise about the operation and use of their application should be able to create fairly robust utility curves that serve the general needs of users. Nevertheless, one of the advantages of our middleware is that it allows this work to be done by usability specialists, without requiring them to tackle complex issues related to network management, as those can be contained completely within the NA.

## 5 Evaluation and implementation

A proof of concept implementation has been built by incorporating the middleware into Marratech Pro [2], a commercially available e-meeting application that provides tools for synchronous interaction including audio, video, chat and a shared white-board. Marratech Pro supports data distribution using IP-multicast or distribution through a media gateway called the Marratech E-Meeting Portal, which can be used when IP-multicast is not available.

The prototype was tested by using two Marratech Pro clients. The first client (client A) used the prototype and was responsible for collecting data during the experiments. The second client (client B) did not adapt bandwidth usage based on the middleware, and was only used in order to change the level of incoming traffic on the link, as this directly affected the available supply as described in the next subsection. Both clients sent video traffic at all times during the experiments, with audio being used by client A at various intervals in order to investigate the effects it had on the system. In the two first experiments, a 100 Mbit local Ethernet network was used to evaluate how the middleware shared bandwidth between media. Client B sent approximately 25 kB/s video traffic in these experiments. In the third experiment, a commercial GPRS network was added to evaluate the BBA. In this case, Client B was configured to only send 3 kB/s video traffic.

Three computers were used during the experiments. The E-Meeting Portal was run on a Pentium III 1.2 GHz machine running Windows XP. Client A was an Intel P4 2.4 GHz machine running Windows XP and Client B was an AMD Athlon 1.2 GHz machine running Windows XP.

### 5.1 Implementation

The prototype was implemented in Java JDK 1.4 in order to make it easy to integrate into the Marratech source code. It followed the middleware as described in section 4 with the agents contained in figure 1 having the following characteristics.

**The Bandwidth Broker Agent** The BBA used the tâtonnement process as described in section 3. In the current implementation it provides an API where different BCAs can register and receive call-backs when the price is updated. The total supply and demand are calculated by using an API provided by the NA, which will be discussed later in this subsection.

**The Bandwidth Consumer Agent** When the price is recalculated each instance of the BCA receives a price update through a call-back. Current demand is calculated based on the price set by the BBA and is used to decide how much bandwidth the BCA should try to purchase. The following utility functions were used for calculating the demand during the experiments. For audio the utility function was

$$u_{audio}(x) = \begin{cases} 0 & x < x_{audio}^{min} \\ \infty & x \geq x_{audio}^{min} \end{cases}$$

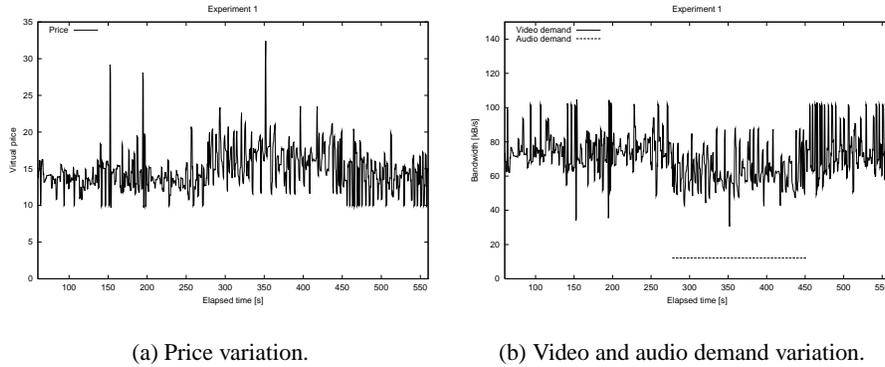
,where  $x_{audio}^{min}$  represents the minimum amount of bandwidth needed by the audio codec. This utility function was used in order to describe the audio media as something very unadaptive, which is the case with many codecs used today, for example GSM. In the experiments, a commercial audio codec called EG711 (GIPS) [1] was used, and  $x_{audio}^{min}$  was set to 12.2 kB/s.

The utility function for the video was modeled using the logarithmic function,  $u_{video}(x) = \ln(1 + x)$ , which was used in order to create a basic concave function. In reality a more complex and accurate function will be more appropriate, but as the purpose of the experiments was to study the marketplace, an optimal utility function was not necessary. Thus, using equation 2, the demand function for bandwidth by the video media is calculated as  $x_{video}(p)^{cs} = \frac{1}{p} - 1$ .

During each price iteration, the BCA informed a bandwidth manager in Marratech Pro about the purchased bandwidth in order to adjust the video encoder to the bits-per-second corresponding to the purchased bandwidth.

**The Network Agent** During the experiments the NA was responsible for calculating the supply. This was done by setting an upper-bound supply limit  $s_{limit}$  for each type of network, and then by calculating the supply  $s$  by subtracting the amount of incoming bandwidth obtained from the operating system from the  $s_{limit}$  (i.e.  $s = s_{limit} - bw_{received}$ ). For the 100Mbit Ethernet network the  $s_{limit}$  was set to 100 kB/s and for the GPRS network it was set to 6 kB/s. Note that this allowed for large fluctuations in available supply by altering the amount of traffic sent out by the other end-point.

Mobility support was provided by using an UDP-socket extension called the Resilient Mobile Socket (RMS) [6]. In practice, it would be possible to use other protocols such as Mobile IP, but RMS was mainly chosen because we already had a working prototype based on RMS.



**Fig. 3.** Results from experiment one. The figures show the effect of introducing a new media into the market. The price was recalculated every 100 ms, and the supply every 1200 ms.

## 5.2 Experiment one: Bandwidth sharing between multiple media

Three experiments using the prototype were conducted. The first experiment was conducted to demonstrate that the prototype could effectively divide the available bandwidth between multiple media. This was done by utilizing all available bandwidth and varying the use of audio at client A in order to show that video would effectively back-off due to price increases in the market.

Figure 3 shows results from this experiment. As shown in figure 3(a), the price goes up almost immediately when audio is sent. This results in a reduction of the demand for bandwidth by the video media, as shown in figure 3(b). This creates the ultimate effect of a reduction in the video bit-rate used by the video encoder, allowing bandwidth to be consumed by the audio encoder.

## 5.3 Experiment two: Investigation into the price and supply recalculation rates

In order to investigate how the price and the supply recalculation rates affected the market, data was collected multiple times while sending video from each client during a period of 10 minutes. The price recalculation rate was studied by locking the supply recalculation rate to 500 ms and decrementing the price recalculation rate from a high value of 1000 ms to a low value of 20 ms. The supply recalculation rate was studied in a similar way with the price recalculation rate locked to 50 ms, instead of by locking the supply recalculation rate.

Table 1 shows the benefits of a higher price recalculation rate, in that it leads to a more efficient allocation of bandwidth, as determined by calculating the average over and under-demand. An explanation is that a higher price recalculation rate improved the response time, allowing the demand to more closely match variations in supply. A high supply recalculation rate on the other hand did not improve the performance as

**Table 1.** The table shows the effects of varying the price recalculation rate. The supply recalculation was recalculated every 500 ms.

Price recalculation rate	Avg. over-demand	Avg. under-demand
20 ms	0.25 kB/s	0.25 kB/s
50 ms	1.14 kB/s	0.87 kB/s
100 ms	2.66 kB/s	2.59 kB/s
200 ms	4.09 kB/s	3.98 kB/s
400 ms	7.18 kB/s	7.10 kB/s
500 ms	7.86 kB/s	7.79 kB/s
800 ms	10.63 kB/s	10.53 kB/s
1000 ms	11.20 kB/s	11.07 kB/s

**Table 2.** The table shows the effects of varying the supply recalculation rate. The price recalculation was recalculated every 50 ms.

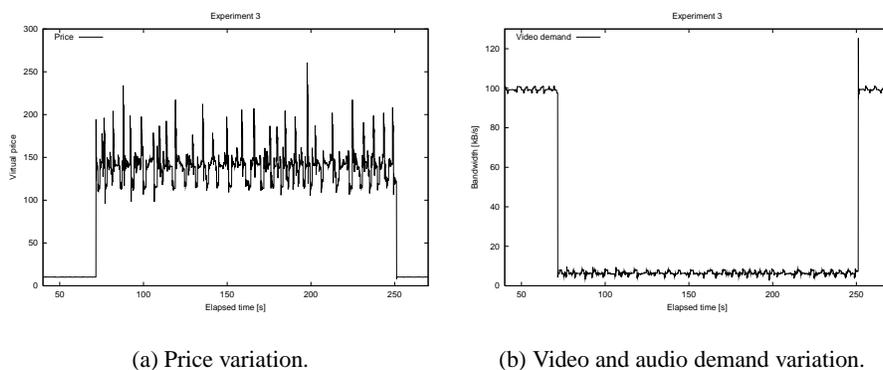
Supply recalculation rate	Avg. demand	Avg. supply	Avg. over-demand	Avg. under-demand
200 ms	97.57 kB/s	86.30 kB/s	11.63 kB/s	4.45 kB/s
400 ms	88.77 kB/s	80.20 kB/s	7.26 kB/s	3.64 kB/s
500 ms	76.51 kB/s	75.20 kB/s	6.87 kB/s	2.10 kB/s
600 ms	74.82 kB/s	74.80 kB/s	1.20 kB/s	1.17 kB/s
800 ms	74.97 kB/s	74.95 kB/s	1.06 kB/s	1.04 kB/s
1000 ms	73.57 kB/s	73.56 kB/s	0.96 kB/s	0.94 kB/s

it resulted in more fluctuation in terms of over and under-demand, which can be seen in table 2. This problem can be explained by the fact that a high supply recalculation rate in combination with variable bit-rate video codecs (H.261) causes supply to vary rapidly, making it harder for the market to reach an equilibrium.

#### 5.4 Experiment three: Obtaining and selling bandwidth from multiple networks

The third experiment was conducted to demonstrate that the BBA could sell bandwidth obtained from more than one network. Another purpose was to investigate how the market reacted when there were large variations in supply caused by mobility. In the experiment, NA was configured to trigger a handover as soon as the LAN interface became available in Windows, and similarly trigger a handover to the GPRS interface if the LAN interface became disconnected. This was done by calling a handover function provided by RMS.

Figure 4 shows the result from the experiments. As can be seen in the figure, the supply dramatically decreases from 100 kB/s to only 6 kB/s after switching to the GPRS network, which consequently caused the price to immediately rise and the video BCA to decrease its demand. Note that the price is less stable on the GPRS network compared with the Ethernet network, which can be explained by the fact that the incoming traffic



**Fig. 4.** Results from experiment three. The figures show the effect on the market when switching between networks with different bandwidth capacity. The price is recalculated every 100 ms and the supply every 600 ms.

(3 kB/s video data) relatively caused more variations in supply on the GPRS network than on the Ethernet network.

## 6 Discussion

This paper has presented a middleware framework based on microeconomic principles of supply and demand that deals with bandwidth issues inside a multimedia application. The key design principle that has been proposed is to view bandwidth as a universal commodity that can be consumed and produced by different components in the application. The advantage of this approach is that the system becomes more modular as each component can contribute to the equilibrium separately in the market. This makes it possible to replace and upgrade each component in the middleware in a “plug and play” style without needing to redesign the whole application. For example, if a new component for mobility management is developed that can take advantage of several wireless base-stations simultaneously [6] it could be integrated into the middleware simply by upgrading the NA. Similarly, if a new method is developed that can better utilize bandwidth in video group communication softwares [15], it can be integrated simply by defining new utility functions in the BCA.

Ultimately, this makes it possible for usability researchers to develop more advanced applications that consume bandwidth in the best possible way without having to care about heterogeneity and complexity in the networks while networking researchers can develop more advanced networking components for obtaining bandwidth without having to consider specific application related issues. Although this is not a new idea in general, we believe that a middleware layer is needed to hide heterogeneity as both applications and networks are becoming more complex to manage.

Moreover, the paper has presented a proof of concept prototype based on the commercially available e-meeting application Marratech Pro. This prototype has been used in several exploratory experiments, which has shown that the middleware can be used in order to share bandwidth effectively between multiple media using the BBA as a single centralized supply point for managing bandwidth.

The experiments have shown that it is possible to allocate bandwidth close to an equilibrium allocation by using a high price recalculation rate and a low supply recalculation rate. However, as a high supply recalculation rate negatively affected the market, studying how a real congestion control scheme affects the performance is something that requires further investigations. Hence, for future work we plan to use a real congestion control scheme and study its implications on the market. In addition, we plan to investigate more effective utility functions for the various media contained in Marratech Pro, and integrate some other related prototypes developed by our research group into the system in order to make more sophisticated experiments.

## References

1. Global IP Sound AB, 2005. Homepage: <http://www.globalipsound.com/>.
2. Marratech AB, 2005. Homepage: <http://www.marratech.com>.
3. E. Amir, S. McCanne, and R.H Katz. Receiver-driven Bandwidth Adaptation for Lightweight Sessions. In *ACM Multimedia*, pages 415–426, 1997.
4. B. Briscoe, V. Darlagiannis, O. Heckman, H. Oliver, V. Siris, D. Songhurst, and B. Stiller. A Market Managed Multi-Service Internet (M3I). *Computer Communications*, 26(4):404–414, 2003.
5. C. Courcoubetis and R. Weber. *Pricing Communication Networks; Economics, Technology and Modelling*. Wiley, 2003.
6. J. Kristiansson and P. Parnes. Application-layer Mobility support for Streaming Real-time Media. In *IEEE Wireless Communications and Networking Conference (WCNC '04)*, 2004.
7. Johan Kristiansson, Jeremiah Scholl, and Peter Parnes. Bridging the Gap between Multimedia Application and Heterogeneous Networks using a Market-based Middleware. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOW-MOM), Extended abstract*, 2005.
8. K. Lai and M. Baker. Measuring Bandwidth. In *INFOCOM*, pages 235–245, 1999.
9. B. Li. *Agilos: A Middleware Control Architecture for Application-Aware Quality of Service Adaptations*. PhD thesis, University of Illinois, USA, 2000.
10. Raymond R.F. Liao and A. T. Campbell. A Utility-Based Approach for Quantitative Adaptation in Wireless Packet Networks. *Wireless Networks*, 7(5):541–557, 2001.
11. J.K. MacKie-Mason and H.R. Varian. Pricing the Internet. In *The second International Conference on Telecommunication Systems Modelling and Analysis*, pages 378–393, 1994.
12. J.K. MacKie-Mason and H.R. Varian. Pricing Congestible Network Resources. *IEEE Journal on Selected Area in Communication*, 13(7):1141–1149, 1995.
13. K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li. QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments. *IEEE Communications Magazine*, 39(11):140–148, 2001.
14. R. Rajkumar, C. Lee, J. Lehoczy, and D. Siewiorek. A Resource Allocation Model for QoS Management. In *Proceedings of the IEEE Real-Time Systems Symposium*, 1997.
15. J. Scholl, S. Elf, and P. Parnes. User-interest Driven Video Adaptation for Collaborative Workspace Applications. In *International Workshop on Networked Group Communication NGC*, pages 3–12, 2003.