Efficient Resource Management for Distributed Applications with Real-Time Requirements in Broadcast Networks

Jürgen Wolf, Bernd E. Wolfinger
Node: University of Hamburg, Germany
email: `jwolf@informatik.uni-hamburg.de`

Abstract

Distributed applications are often established in local, e.g. wireless, networks. An important means to guarantee an acceptable quality of service in networks with real-time communication requirements is the reservation of resources at connection setup time. However, such reserved resources, e.g. transmission bandwidth, may be unused as a consequence of the variations in the actual resource demands. Indeed, in many, mainly wireless, networks these resources are quite limited. Therefore, we propose a more efficient resource utilization which is possible if communicating stations or end-users dynamically hand over some of the free resources temporarily to the other communication partners, e.g. of a "broadcast network".

This paper summarizes some of our main results as achieved up to now. We focus on our recent work in progress as well as on future prospects of such a demand-based sharing of resources: on the one hand, estimation of the current resource requirement on the basis of load measurements is discussed and, on the other hand, we present efficient algorithms for resource sharing respecting real-time requirements. Our approach suggested for resource and traffic management allows one to achieve significantly better utilization of network resources which is demonstrated by means of examples.

1 Introduction

The trend towards distributed multimedia communications via local-area networks has led to the requirement to transmit continuous media streams, such as audio and video streams, with sufficiently good quality. One common possibility in order to guarantee a certain QoS as required by the users of a network is *resource reservation* as reflected e.g. by the IntServ proposal [BCS94]. In this paper we focus on extending this approach and its usage in local broadcast networks. In particular, we study ways of how communication resources which are already reserved and allocated to communicating end-users or end-systems can be dynamically handed over to other end-users or end-systems in case that they are not needed by their original "owner" for some time. Our interests center around methods which assure that resources are recalled sufficiently quickly, when they are needed again by their owner, in order to make sure that real-time requirements are not neglected.

The approach we suggest assumes that the problem of reserving and statically allocating resources to communicating entities has already been solved [ChG02] and that the resource reservation has been established for a rather long-term time interval, e.g. duration of an audio/video stream. Each owner of resources determines whether its communication load justifies the continued reservation of all the resources as allocated to himself. If a sufficiently large amount of resources is observed to be temporarily free, the owner will pass some of these resources to its "neighbors". In order to respect real-time requirements for its data to be sent each owner will continue to observe the arrivals to its transmission queue. If its local load is increasing again, the owner informs its neighbors that they are no longer allowed to make use of the resources which were offered for public access at an earlier instant. The scenario our approach assumes is introduced in more detail in Section 2 and in [WWL03]. The freeing of resources and their reclaiming could lead to oscillations why our approach is based on load estimators, as presented in Section 3, which produce some kind of smoothed estimates for the load as it is generated locally over time. Moreover, in Section 4, we introduce a system of load thresholds which, only when being crossed by the load estimates of an owner of resources, leads to a broadcast of messages for freeing or reclaiming resources. Section 5 shows results for three sample threshold systems in combination with different load estimation functions. In Section 6 we summarize the current state of our work and indicate future prospects.

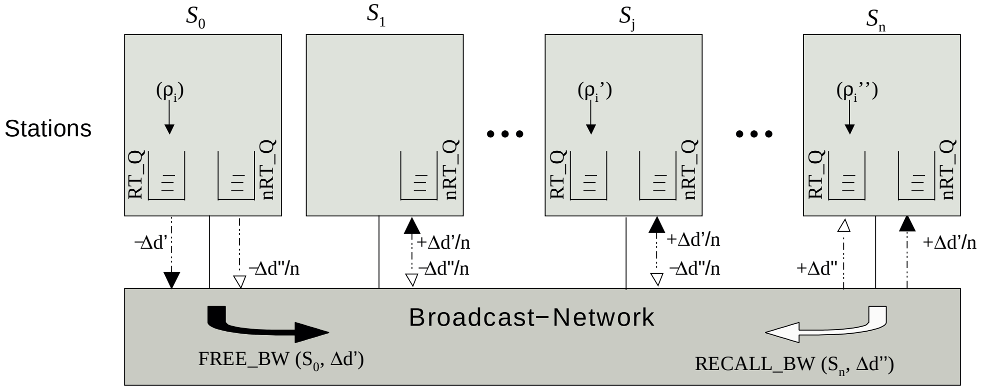

Figure 1: Redistribution of bandwidth: the depicted example scenario illustrates the freeing of $\Delta d'$ by station S_0 and recalling of $\Delta d''$ by S_n

2 Underlying model for resource distribution

In the following we present the basic proceeding which we suppose throughout this paper for resource reservation and redistribution between the stations of a broadcast network. The assumptions and the underlying model for resource redistribution are as follows:

- Resource reservation:
  We assume:
  - $n + 1$ stations $S_0, \ldots, S_n$ communicating via a local broadcast network (Ethernet, WLAN,…[Tan03])
  - communication load of the stations consisting of time-critical transmissions and non-time-critical data transfers, cf. the real-time (RT_Q) and non-real-time queues (nRT_Q) in Figure 1
  - in each station with real-time communication requirements there exists at least one owner of resources, where reserved data rate (allocated "bandwidth") is the resource considered by way of example; scenarios could be that the owner reflects the complete station or an owner could model a single end-user
  - resources are supposed to be reserved before starting a new stream with real-time requirements and are available to its owner until it releases connection after end of the stream; the amount of bandwidth reserved is such that the owner's QoS requirements can be guaranteed.
- Temporary resource redistribution:

  We assume that an owner of resources can pass resources to other stations temporarily, whenever these resources are not required locally for some time. We introduce a message of type

  - FREE_BW $(S, \Delta d)$ with which a station S hands over a bandwidth (respectively data rate) of Δd bit/s to other stations, and
  - RECALL_BW $(S, \Delta d)$ to recall bandwidth Δd.

  With receipt of a RECALL_BW message the recalled bandwidth has to be released immediately. The question of how to distribute free bandwidth among stations is out of the scope of this paper; an elementary solution could be, that all neighboring stations get equal shares. A more advanced approach to distribute free bandwidth efficiently is proposed in [AnL01].

- Approach to estimate the level of load resulting from time-critical transmission requests:

  We assume periodic measurements available at instants $t_i := t_0 + i \cdot \Delta t$, for $i \geq 1$, where the measured samples ρ_i characterize the load of time-critical transmission requests which have been generated by the source during the interval $T_i := [t_{i-1}, t_i)$. Evidently, $\rho_i := d_i/(r \cdot \Delta t)$, where d_i denotes the amount of time-critical data generated during T_i and r denotes the data rate as it was allocated to its owner at connection set-up time. This sequence of samples (ρ_i) may be used at each instant t_i to calculate an estimate $\hat{\rho}(t_i)$ of the actual level of load.

3 Estimation of actual node utilizations using geometric and arithmetic weighting

The traffic load generated by the communicating users of a station must be estimated in order to be able to adjust the allocated capacity to the actual requirement. If the capacity, in terms of transmission bandwidth, allocated is too high, only a poor level of utilization for the network can be achieved. Some flows may have more reserved capacity than what they really use. This capacity is lost for all the nodes of the network. On the other hand, if the reserved capacity is too low in comparison to the flow activity, a situation of congestion occurs and the general QoS of the flow will be degraded. It is important to find out good estimators to allow the algorithm controlling the bandwidth to react in an appropriate manner.

The estimator must take into account some kinds of fluctuations in order to enable adequate node reactions. Major changes in data rates for a complex source of traffic must be detected quickly. This is an important condition for flows with a required level of QoS. If the reaction is too slow, a QoS degradation is viewed by these flows, whereas a minor change in data rates within single streams may be ignored in order to get a certain form of smoothness over time. This constraint is to avoid an overreaction with respect to the observed fluctuation of offered load within a station.

The methods proposed and investigated by us for load estimation are *geometric* and *arithmetic* weighting.

Here, ***geometric weighting*** is defined as follows:

$$\hat{\rho}(t_i) = \alpha\rho_i + (1-\alpha)\cdot\hat{\rho}(t_{i-1}), \quad i \geq 1$$

where $\alpha \in (0,1]$ and $\hat{\rho}(t_0)$ to be initialized. The α factor indicates how fast the estimator changes with a strongly different new sample. The geometric weighting is also known as the *exponential weighted moving average* (EWMA) [Rat91]. As an abbreviation we denote by G_α geometric weighting with parameter $\alpha \in (0,1]$.

The ***arithmetic weighting*** estimator is defined as:

$$\hat{\rho}(t_i) = C_0 \cdot \sum_{j=0}^{w-1} \frac{w-j}{w}\rho_{i-j}, \quad i \geq 1,$$

where $w \in \{1,2,\ldots\}$ and ρ_k to be initialized for all $2-w \leq k \leq 0$. $C_0 = \frac{2}{w+1}$ denotes a normalization constant. This estimator is equivalent to a sliding window with window size w and weights increasing in a linear way when the samples are more recent. Arithmetic weighting uses $\frac{k}{w}$ as weights, where k is the position in the window of size w with the most recent sample taking the position $k = w$. The size of w indicates how many past samples including the present sample must be kept to estimate the current load. As an abbreviation, A_w denotes arithmetic weighting with a window size of $w \in \mathbb{N}$.

4 System of load thresholds

In this section we introduce a system of load thresholds, which allows to determine when to free or claim back bandwidth based on current load estimation as discussed in the preceding section. The main tasks of such a system are on the one hand to assure real-time requirements to be respected and secondly to avert oscillations caused by small variations of the estimator's value.

Definition 1 *We define an* ***n*-state threshold system** *as tuple* $TS(S,\vartheta)$. *The* n*-tuple* $S = (S_1, S_2, \ldots, S_n) \subset [0,1]^n$, *with* $S_i < S_j$ *for all* $1 \leq i < j \leq n$, *denotes the set of states while the state-transitions are defined by the* $n \times (n-1)$*-matrix*

$$\vartheta = \begin{pmatrix} \vartheta_{1,2} & \cdots & \vartheta_{1,n} \\ \vdots & \ddots & \vdots \\ \vartheta_{n,2} & \cdots & \vartheta_{n,n} \end{pmatrix} \subset (0,1]^n \times (0,1]^{n-1}$$

with $i < j \Rightarrow \vartheta_{k,i} < \vartheta_{k,j} \quad \forall k$. *With given state* S_i *and estimated load* $\hat{\rho}$ *the next state is given by*

$$r(S_i, \hat{\rho}) := S_{\max\limits_{j\in\{1,..,n\}}\{\vartheta_{i,j}\leq\hat{\rho}\,,\ 1\}}$$

Interpretation of ϑ:

- Up-thresholds: For $i < j$, each entry $\vartheta_{i,j}$, determines the up-threshold which has to be crossed upwards (or at least reached) by the estimator $\hat{\rho}$, in order to change the system's state from S_i to S_j.
- Down-thresholds: For $i \geq j$, each entry $\vartheta_{i,j}$, determines the down-threshold which has to be crossed downwards by the estimator $\hat{\rho}$, in order to change the system's state from S_i to S_{j-1}.

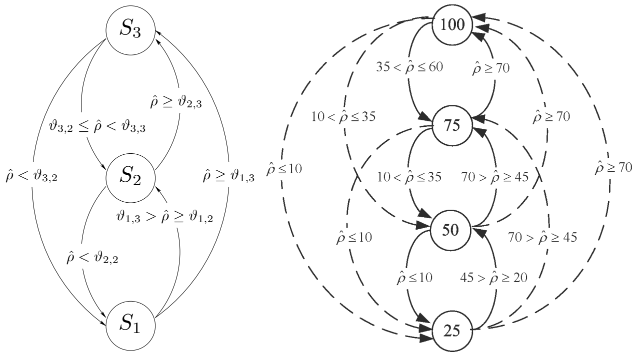

Figure 2: A general 3-state threshold system (left picture) and M2, an example 4-state threshold system; states and transitions of M2 are labeled in percent.

If neither any up-threshold nor any down-threshold is crossed, the system remains in the current state. Figure 2 shows a general 3-state (left picture) and a specific, i.e. actually parameterized, 4-state threshold system by way of example.

In contrast to assessments regarding the gained bandwidth by means of resource redistribution, we currently can not give a closed-form formula to calculate the maximum delay, though it can be determined recursively [WWL03]. Currently, we are also investigating how to manage automatic parameterization of the system for a given application and its real-time requirements. As a final goal we strive to solve the optimum configuring of adaptive reservation systems, for given load scenarios and given QoS requirements.

5 Sample configurations of threshold systems

The main goal is to evaluate how well applications' real-time constraints can be guaranteed by using the proposed threshold systems in combination with different load estimation functions. In this section we present results obtained for some sample configurations. We define three models $M1, M2$, and $M3$ as 2-state, 4-state, and 9-state threshold system respectively, $M2$ is shown in the right picture of Figure 2 by way of example.

$$M1 = TS\left((0.5, 1); \begin{pmatrix} 0.30 \\ 0.45 \end{pmatrix}\right)$$

$$M2 = TS\left((0.25, 0.5, 0.75, 1); \begin{pmatrix} 0.20 & 0.45 & 0.70 \\ 0.10 & 0.45 & 0.70 \\ 0.10 & 0.35 & 0.70 \\ 0.10 & 0.35 & 0.60 \end{pmatrix}\right)$$

$$M3 = TS\left((0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1); \begin{pmatrix} 0.15 & 0.25 & 0.35 & 0.45 & 0.55 & 0.65 & 0.75 & 0.85 \\ 0.10 & 0.25 & 0.35 & 0.45 & 0.55 & 0.65 & 0.75 & 0.85 \\ 0.10 & 0.20 & 0.35 & 0.45 & 0.55 & 0.65 & 0.75 & 0.85 \\ & & & \vdots & & & & \\ 0.10 & 0.20 & 0.30 & 0.40 & 0.50 & 0.60 & 0.70 & 0.80 \end{pmatrix}\right)$$

We studied node and network behavior for several real-time streams. One typical example is the stream that originates from the two superimposed MPEG-streams of the movies Jurassic Park and Mr. Bean [Wol00]. This stream lasts about 60 minutes and we chose an appropriate value for the interval length Δt between the load measurements, namely $\Delta t = 40ms$. We then calculated the behavior for different levels of reserved transmission capacity. The left diagram in Figure 3 shows the share of the reserved bandwidth which could be utilized by the overall network by means of adaptive resource redistribution with different load estimators and with our three models as chosen by way of example. The white bar indicates how much of the reserved resources are utilized by the video stream itself. The additional resource utilization is gained by the resource redistribution system. It can be seen that model M3, for example, makes it possible for the network to use virtually 90% of this stream's reserved resources though the stream itself uses only about 30%. In this study we neglect the effects of the signaling procedures. Nevertheless, the average rate of messages sent for freeing and recalling bandwidth are depicted by the squares in the left diagram of Figure 3.

As drawback, the resource redistribution induces additional delays. These delays depend on the choice of load estimator and threshold system. The bars in the right diagram of Figure 3 indicate the maximum additional delays in milliseconds that occur during the whole 60 minutes stream. Furthermore the mean delays are given with their 95% confidence intervals. At least for models M1 and M2 the confidence intervals are quite narrow, which indicates that large delays appear rarely and that delays are expected to be in the range of about 5 - 15ms. That is what most applications could tolerate even when once in a while larger delays would appear. Of course the level of tolerance depends on the individual application. Therefore, it is an important task for further studies to get still better understanding regarding the dependencies between the parameterization and to provide mechanisms for adopting these parameters to the needs of applications.

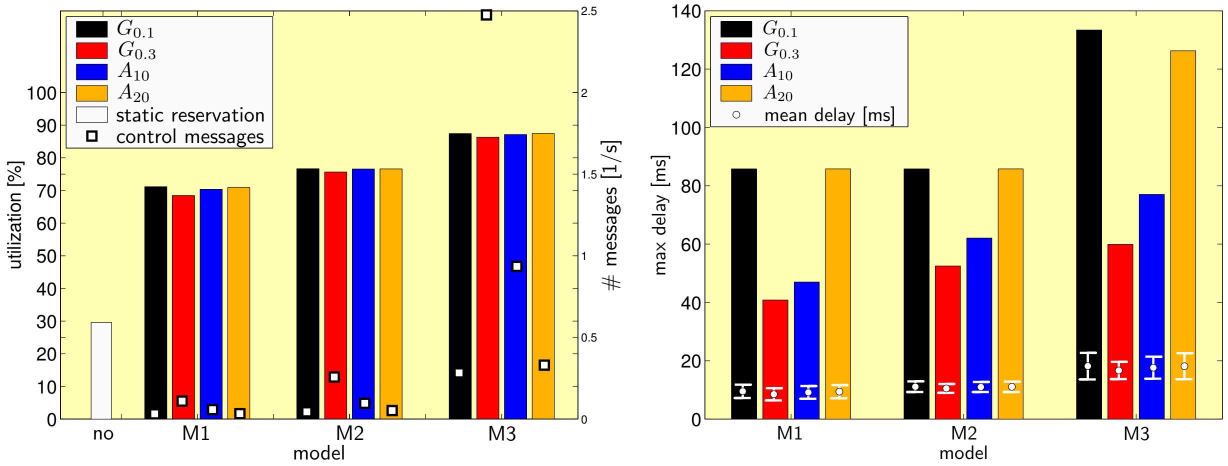

Figure 3: Video mix sent with $r = 4.4\ Mbit/s$ using different combinations of estimators and threshold systems: left diagram: overall bandwidth available for usage within the network; right diagram: maximum and mean delay (including 95% confidence intervals)

6 Summary and future work

In this paper, we have highlighted how the efficiency of reservation strategies that execute some fixed resource reservations can be increased, in particular with regard to distributed applications. On the one hand, we have proposed two classes of parameterized weight functions for calculating estimates of the actual load as induced by the connections for which resource reservations have been established for the present. On the other hand, we have introduced a system of load thresholds in order to reduce the update traffic for exchanging control information the purpose of which is to hand over free bandwidth and to recall own resources between the stations of a broadcast network. We have demonstrated how weight functions and threshold systems can be combined. Additional studies have provided the encouraging result that in practice, i.e. for realistic load scenarios, the efficiency to be expected for our proposal to execute dynamic and load-adaptive resource reservations will significantly decrease the sacrifice of a priori reserved capacity.

One important task for further studies is to get still better understanding regarding the dependencies between the parameterization and to provide mechanisms for adopting these parameters to the needs of applications, at best in a largely automatic way. Moreover, we are currently working on a simulation prototype for additional analyses of overall network behavior, in particular to investigate wireless communication scenarios regarding distributed applications. Also, we have planned to extend the study to alternate workload estimators, e.g., as an alternative to observing utilization, one could observe the current backlog and base resource redistribution on this estimate.

References

[AnL01] Anelli, P., Le Grand, G., Differentiated Services over Shared Media, In *IWQoS*, pages 288–293, Karlsruhe, 2001

[BCS94] Braden, R., Clark, D., Shenker, S., Integrated Services in the Internet Architecture: An Overview, RFC 1633, IETF, June 1994

[ChG02] Chao, H. J., Guo, X., *Quality of Service Control in High-Speed Networks*, J. Wiley, 2002

[Rat91] Rathgeb, E. P., Modelling and Performance Comparison of Policing Mechanisms for ATM Networks, *IEEE Journal On Selected Areas In Comm.*, 9(3):325–334, 1991

[Tan03] Tanenbaum, A. S., *Computer Networks*, Prentice-Hall, 4th edition, 2003

[Wol00] Wolisz, A., MPEG-4 and H.263 Video Traces for Network Performance Evaluation, Available under http://www-tkn.ee.tu-berlin.de/research/trace/trace.html, 2000

[WWL03] Wolf, J., Wolfinger, B., Le Grand, G., Improving Node Behavior in a QoS Control Environment for Local Broadcast Networks, In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, Montreal, 2003