Synchronized Access Networks*

Christof Brandauer, Peter Dorfinger, Vinod Kone
SalzburgResearch
Jakob-Haringer-Str. 5/III
A-5020 Salzburg, Austria
{brandauer,dorfinger,vkone}@salzburgresearch.at

Abstract

This paper discusses an approach for coupling local real-time networks over an IP core network. The proposed service class provides deterministic guarantees on delay and jitter. To realize this, synchronized transmission schedules are employed in the access areas of the network. The schedule precludes resource contention among the flows and enables a conflict free transmission at the IP layer. A mathematical model for the request admission probability is derived for a simple allocation scheme. First simulation results show that this probability can be increased significantly with alternative allocation algorithms.

1 Introduction

The work presented in this paper is motivated mainly by two observations. The first one is that embedded systems are becoming ubiquitous. Many of the products using embedded systems need to communicate with other embedded systems. Primarily this is realized on local communication systems that often provide deterministic real-time services (e.g. production plants).

The second observation is that the Internet Protocol IP [11] is settling as *the* standard convergence layer in wide-area packet networks. While there is a big variety of higher layer protocols on the one side and subnetworking techniques on the other side, IP is almost exclusively the common denominator at the network layer.

Given these tendencies we believe there is an application for coupling local real-time networks over an IP network providing deterministic guarantees. In this paper we present an idea for realizing such a service in an environment where the local real-time networks are available (e.g. Profiline, Powerlink, etc). It is not the goal to use the service to ”make the whole Internet real-time”.

The objective is to create a class that provides - in this order - 1) a deterministic upper bound on delay and delay jitter and 2) a low delay and jitter. It must be based on standard IP without any modification. The new approach taken is to investigate the concept of synchronized transmission schedules at the IP layer. Transmission schedules are computed for the delay critical parts of a path such that a conflict free flow of packets is established in these regions. For easier reference, the envisaged service class is referred to as *SA service* (Synchronized Access) below.

*This work is funded by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT contract FFG 807142

2 Approach

The indeterministic nature of network delays is to a large degree determined by the stochastic queueing delays. Propagation and transmission delays (assuming a maximum packet size) are deterministic. In today’s typically highly overprovisioned IP core networks the queueing delay is insignificant. It mostly occurs in the access networks that are characterized by low to medium bandwidth links and the concentrator functionality. When several flows are multiplexed on a medium bandwidth link, queueing delays can quickly become high. As an example, consider a 2 Mbit/s link where 10 flows are multiplexed. Each packet shall have a size of 10000 bit. Even if it is assumed that each flow has never more than 1 packet at the concentrator, the worst case delay is already 50 ms for that hop. It is obvious that a worst case end-to-end delay bound will be too high for many real-time applications.

Taking into account the different characteristics of the network regions we employ specific strategies to construct the SA class. The main focus is on the access networks because they are i) the primary source of delay unpredictability and have ii) the largest potential for delay reduction.

The network is logically divided into access and core region. An access router (AR) resides in the access network and connects (a) local network(s) with a domain’s edge router which has the connection into the core of the network.

2.1 Core network

The core network is characterized by high bandwidth and low link utilization. This is a typical situation in today’s wide-area backbone networks. The IP core could also be a ‘local’ IP backbone connecting a number of LANs, e.g. in a large production plant. Due to the core characteristics the queueing delay can be expected to be small. There is no persistent congestion, queues can only build up due to short-time traffic bursts. Additionally, the SA service will be

handled in a separate traffic class that is granted high priority access to the link. Admission to this service is limited. These conditions provide for very little queueing delay. To make it a deterministic component we propose to compute a worst case queueing delay. Given the small delays, even a worst case computation of the delay should yield a sufficiently small value. We expect that a more sophisticated and fine-grained delay computation would not result in a significantly smaller upper bound for the queueing delay. We do therefore not further consider this issue for the moment.

2.2 Access network

The large potential for the reduction of delays is in the access network, on the ingress as well as on the egress side. The goal is not only to provide a deterministic but also a small upper bound on network delay.

In order to reduce queueing delays in the access area of the network we propose a time-triggered synchronization of traffic at the IP layer. Like in a classical TDMA approach, time is conceptually divided into time slots. A number of slots are logically combined to a frame. For a given set of requests a transmission schedule is computed. The schedule covers one complete frame and is always repeated with the beginning of the next frame. The schedule is computed such that no more than 1 packet per time slot has to be sent at router's output port. If such a schedule exists, a conflict free transmission of IP packets is guaranteed for that router. Packets will always find an empty queue at the router output port and will thus experience no queueing delay. This concept makes the packet forwarding a deterministic task as the concurrent competition for bandwidth is precluded.

On the ingress side, the transmission schedule is computed for the link from the access to the edge router because this link is considered the bottleneck on the ingress side. Concerning the connection between the end-systems and the access router it is assumed that some local real-time network enables an end-system to deliver a packet at a specified times. These times are allocated by a resource manager after a service request was accepted.

2.3 Synchronization of ingress and egress

Analogous to the ingress side, a conflict free transmission schedule is employed for the link from the egress edge router to the egress access router.

Between the ingress access router and the egress edge router the worst case queueing delay through the core network is known. It is therefore known when packets are ready to be sent at the egress edge router. If these time slots are indeed allocated for that flow we call this a zero delay (ZD) allocation scheme. Packets that arrive early are buffered at the egress edge router. It must be guaranteed that early packets from one flow can not delay packets from other flows.

If a ZD scheme is not feasible because the requested time slots are occupied it is possible to exploit the delay budget (if any) that is given as the user's requested delay minus the worst case delay through the core network. This budget can be used to increase the probability of admission.

One possibility is to delay each packet of a flow by a constant number of slots at the egress router. We denote this scheme as constant delay egress, short CDE. Another variant is to selectively allocate slots with a variable delay at the egress router (VDE). One can easily construct realistic request/release sequences where a new flow can only be admitted if this variable delay allocation scheme is used at the egress. Finally, the maximum utilization is reached if allocation delays are jointly exploited at the ingress and egress router (VD). The VDE and VD scheme can only be used with respect to jitter constraint specified in the service request.

If the transmission schedule on the ingress side is established independently of the egress side, the send times of packets at the egress ER are fixed (worst case delay through the core). There is thus no flexibility in trying to accommodate a new request. It can only be checked whether the request fits in or not.

If, however, the schedule for the ingress AR and egress ER is searched for collaboratively, the resource utilization gets higher as the free time slots can be matched to one another. To do this, a *merged* frame is created by aligning the egress frame to the ingress frame (shift by worst case delay) and logically combining them: a slot in the merged frame is free if and only if it is free in the ingress and aligned egress frame at that position.

3 Probability of admission

In this section we derive the probability that a request can be accepted in the ZD scheme. The frame class $F_{N,s}$ is defined as the set of frames that have a length of N slots out of which s slots are free. The slot positions within a frame are numbered from 1 to N, a slot is either in state `free` or `busy`. The function $S : \{1, \ldots, N\} \mapsto \{\texttt{free},\texttt{busy}\}$ maps a slot position to its state. It is assumed that the probability that a slot is free is the same for all positions. Each frame $\in F_{N,s}$ has a distinct set of free slot positions.

A set of indices $A_{N,s,f} = \{p_1, \ldots, p_f\}$, $p_i \in \{1, \ldots, N\}, p_i < p_{i+1}$ is defined as an *allocation* for a frame $\in F_{N,s}$ if the following conditions are fulfilled:
$\cdot\ p_{i+1} - p_i = N/f\ , \forall\, i \in \{1, \ldots, f-1\}$ and
$\cdot\ S(p) = \texttt{free}\ , \forall\, p \in A.$
A frame is said to *contain* an allocation A if
$S(p) = \texttt{free}\ , \forall p \in A.$

3.1 Single frame

First we derive the probability that a request with frequency f can be accepted in a frame $\in F_{N,s}$. The acceptance probability $P_A(N, s, f) = X/Y$ where X equals the number of frames $\in F_{N,s}$ that contain at least one feasible allocation and Y equals the total number of frames $\in F_{N,s}$ which is given by $\binom{N}{s}$.

Note that a frame can contain multiple allocations, e.g. a frame with all slots empty contains all feasible allocations. It must be ensured that no frame is counted more than once. The term X can be calculated by applying the Principle of Inclusion and Exclusion (PIE). Let $n = N/f$. We divide the N slots into f groups of n slots each. Each position p from

an allocation A must be in a distinct group. There are n allocations. The number of frames $\in F_{N,s}$ that contain exactly a allocations is given by $\begin{pmatrix} n \\ a \end{pmatrix} \times \begin{pmatrix} N-(a\times f) \\ s-(a\times f) \end{pmatrix}$.

For each frame containing an allocation, the first slot of the allocation can be chosen in $\binom{n}{a}$ ways (the remaining $f-1$ slots of the allocation are fixed by the first) and the remaining $s-af$ free slots can be chosen in $\binom{N-af}{s-af}$ ways.

As each allocation requires f slots, the maximum number of allocations that a frame can contain is $m = \lfloor \frac{s}{f} \rfloor$.

For simplicity, we define
$g(i) = \begin{pmatrix} n \\ i \end{pmatrix}$ and $h(i) = \begin{pmatrix} N-(i\times f) \\ s-(i\times f) \end{pmatrix}$.

By application of PIE, the number of frames $\in F_{N,s}$ that contain at least one feasible allocation is thus given by:

$$\begin{aligned} X &= g(1)h(1) - g(2)h(2)\ldots(-1)^{m+1}g(m)h(m) \\ &= \sum_{i=1}^{m}(-1)^{i+1}g(i)h(i) \end{aligned}$$

In total, the probability that a request with frequency f can be accepted at a frame of length N with s slots free is given in equation 1.

$$P_A(N,s,f) = \frac{\sum_{i=1}^{m}(-1)^{i+1}g(i)h(i)}{\binom{N}{s}} \qquad (1)$$

3.2 Zero Delay

It is assumed here for simplicity that both frames have the same length N and the same number of free slots s. In each frame, the free slots are assumed to be at random positions. The creation of the merged frame as described in section 2.3 results in a new frame $\in \{F_{N,s^*}\}$. The number of free slots s^* in the merged frame depends on the positions of the free slots in the original frames. Clearly, $\max(0, 2s-N) \leq s^* \leq s$. The probability P_M that the merged frame contains exactly s^* slots is given by:

$$P_M(N,s,s^*) = \frac{\binom{s}{s^*}\binom{N-s}{s-s^*}}{\binom{N}{s}} \qquad (2)$$

As the frame classes F_{N,s^*} are a partition of the sample space the total probability theorem can be applied to compute the probability P_{ZD} that the request can be accepted under the ZD scheme:

$$P_{ZD}(N,s,f) = \sum_{s^*=s_0^*}^{s} P_M(N,s,s^*)P_A(N,s^*,f) \qquad (3)$$

where $s_0^* = \max(0, 2s-N)$.

Figures 1 and 2 show plots for 2 sample scenarios with $N = 32$ and $N = 64$, respectively. Different values for s were used. Note that P was only calculated for specific frequencies (the points in the plot) and the connecting lines are just made to enable easier mapping of the points to their s value.

As can be seen, the acceptence probability P decreases rapidly with an increasing request frequency. The real-time

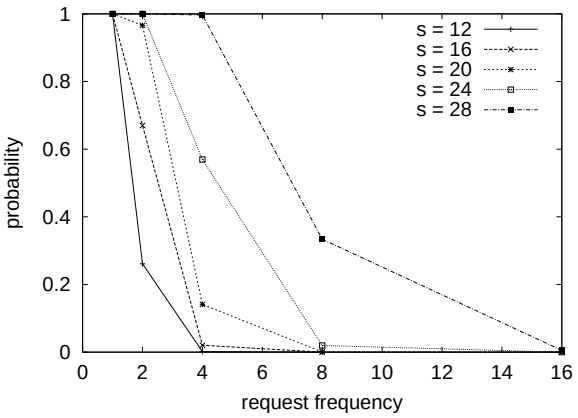

Figure 1. Admission probability in the ZD scheme, N = 32

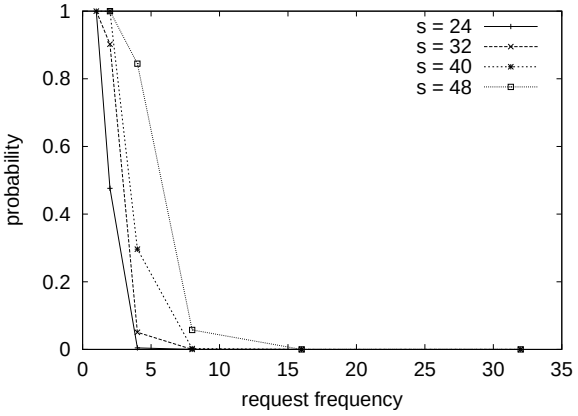

Figure 2. Admission probability in the ZD scheme, N = 64

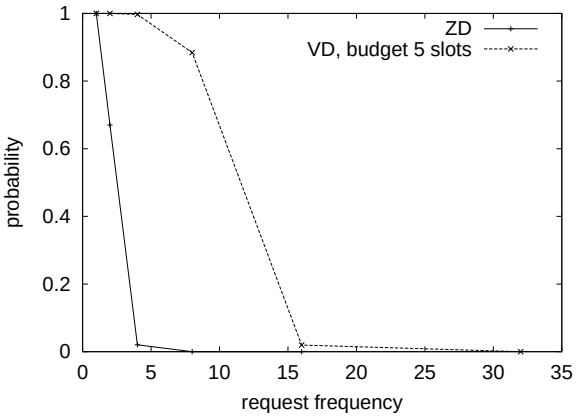

Figure 3. Comparison of admission probability between ZD and VD scheme, N = 32, s=16

traffic can thus only utilize a small portion of the available capacity. Although the remaining capacity can be allocated to other elevated classes and best effort traffic, a higher admission probability as can be achieved by the CDE, VDE, and VD scheme, is clearly desireable.

We do not yet have analytical models for the variable delay schemes. We made a Java implementation of the resource management layer with the different allocation schemes. This implementation was thus far used as a simulation tool but most of the components will later be reused in a laboratory testbed.

For the simulation, random requests are generated and submitted to a resource manager. To compare the simulation with the model, random positions for the free slots are chosen each time before it is tried to accommodate the request. The positions of free slots are chosen independently for ingress and egress, however the number of free slots is the same.

Simulations, where only the ZD allocation scheme is enabled, show a perfect match with the admission probabilities given by the model in equation 3.

Figure 3 compares the admission probability between the ZD and VD scheme for the case of $N = 32, s = 16$. The VD scheme was given a budget of 5 slots for each request. It can be seen that in this case the admission probability is significantly increased for frequencies 2, 4, and 8.

4 Related work

To implement a synchronized transmission schedule the participating nodes must have synchronized clocks. A lot of work has been done in the area of synchronizing physically dispersed clocks over packet networks, see [13, 1] for an overview.

We plan to follow an architecture that is similar to the hierarchical CesiumSpray approach [14]. There, GPS clocks are used for the task of externally synchronizing distributed nodes. Each of these nodes in turn distributes the highly accurate official time as a reference time into a local network to which it is attached. In the architecture shown in this paper, access routers would be the right devices to host the GPS receivers. For internal synchronization with end-systems and edge routers, the access router distributes the GPS time to the attached devices.

For internal synchronization (possibly via Ethernet [10]) the IEEE 1588 protocol [4] could be used. It seems to become the de facto standard for high precision clock synchronization in industrial environments. With additional hardware support SynUTC [12, 7] can achieve even higher precision.

The central function of the admission control module is the computation of a feasible transmission schedule. A wealth of literature on scheduling mechanisms is available, see [9] for a comprehensive overview. Most scheduling problems in the real-time literature are based on the assumption that the whole set of requests is known in advance [8, 2, 3, 5, 6]. For the SA service a different approach is needed because the number of active flows varies over time. The end-systems can request admission during the runtime of the network. Admission control must decide online whether the new request can be accommodated along the active flows.

5 Conclusion

This paper presents an approach for an IP service class that can be used to couple existing local real-time networks while keeping deterministic delay and jitter guarantees. The idea is to employ synchronized transmission schedules in the delay critical access areas of a network. The concept is illustrated and a model for the admission probability under a zero delay allocation scheme is derived. It is shown that this scheme is able to utilize only a very small portion of the available capacity. First simulation results show, as expected, that the admission probability can be significantly increased with a variable delay allocation scheme.

References

[1] E. Anceaume and I. Puaut. A taxonomy of clock synchronization algorithms, 1997.

[2] L. Dong, R. Melhem, and D. Mosse. Time Slot Allocation for Real-Time Messages with Negotiable Distance Constrained Requirements. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium*, 1998.

[3] L. Dong, R. Melhem, and D. Mosse. Scheduling Algorithms for Dynamic Message Streams with Distance Constraints in TDMA protocol. In *Proceedings of the Euromicro Conference on Real-Time Systems*, 2000.

[4] J. C. Eidson, M. C. Fischer, and J. White. IEEE-1588 standard for a precision clock synchronization protocol for networked measurement and control systems. In *34 th Annual Precise Time and Time Interval (PTTI) Meeting*, pages 243–254, 2002.

[5] C. Han. Scheduling real-time computations with temporal distance and separation constraints and with extended deadlines. Technical report, University of Illinois at Urbana-Champaign, 1992.

[6] C. Han, K. Lin, , and C. Hou. Distance-Constrained Scheduling and Its Applicatios to Real-Time Systems. *IEEE Transactions on Computers*, 1996.

[7] R. Höller, M. Horauer, G. Gridling, N. Kerö, U. Schmid, and K. Schossmaier. SynUTC - High Precision Time Synchronization over Ethernet Networks. In *Proceedings of the 8th Workshop on Electronics for LHC Experiments*, pages 428–432, Colmar, France, September 9-13 2002.

[8] R. Holte, A. Mok, L. Rossier, I. Tulchinsky, and D. Varvel. The Pinwheel: A real-Time Scheduling Problem. In *Proceedings of the 22nd Hawaii International Conference on System Science*, 1989.

[9] Joseph Y-T. Leung and James H. Anderson. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman & Hall/CRC, April 2004. ISBN 1584883979.

[10] Institute of Electrical and Electronics Engineers. IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2002. IEEE Std. 802.3-2002.

[11] J. Postel. RFC 791: Internet Protocol, September 1981.

[12] Ulrich Schmid and Klaus Schossmaier. Interval-based clock synchronization. *Journal of Real-Time Systems*, 12(2):173–228, 1997.

[13] B. Simons, J. Welch, and N. Lynch. An overview of clock synchronization. In *Fault-Tolerant Distributed Computing*, volume volume 448 of LNCS, pages 84–96, 1990.

[14] P. Verissimo, L. Rodrigues, and A. Casimiro. CesiumSpray: a Precise and Accurate Global Clock Service for Large-scale System. *Journal of Real-Time Systems*, 12(3):241–294, May 1997.