STUDIA INFORMATICA 2002
Volume 23 Number 2A (48)

Jerzy MARTYNA
Jagiellonian University, Dept. of Computer Science

SUPPORTING HARD REAL-TIME APPLICATIONS IN ATM NETWORKS - NEXT GENERATION LANS

Summary. In this paper, we consider the problem of applying the ATM technology for building distributed industrial plant control applications with hard real-time constraints. We propose a hard real-time client-server programming model implemented in an ATM network. Next, we present appropriated traffic models and some mathematical implications of the used considerations. Finally, we formulate new metrics for evaluating the scheduler on switches needed for the implementation of hard real-time applications in ATM networks.

WSPOMAGANIE APLIKACJI NIEPRZEKRACZALNEGO CZASU KRYTYCZNEGO W SIECIACH ATM – NOWA GENERACJA LOKALNYCH SIECI KOMPUTEROWYCH

Streszczenie. W pracy przedstawiono zagadnienie zastosowania technologii ATM do realizacji aplikacji nieprzekraczalnego czasu krytycznego. Zaproponowano model wykonywania aplikacji typu klient-serwer, który może być zaimplementowany w sieci ATM. Dla zagwarantowania ograniczeń nieprzekraczalnego czasu krytycznego wprowadzono nowy algorytm szeregowania połączeń w sieci ATM, którego obliczone parametry zostały porównane z innymi. Wprowadzono nowe miary oceny połączeń czasu krytycznego w sieci ATM.

1. Introduction

In the last few decades, our desire to carry video and digital audio over packet-switched networks has motivated the development of the ATM technology. It is obvious that it is an excellent technology to provide integrated services for high-speed digital communication networks. Its ability to support high bandwidth, scalability, and guaranteed Quality-of-Service

(QoS) parameters make the ATM an ideal network for distributed applications with diverse statistical characteristics and QoS requirements. However, the use of the ATM technology to support hard real-time communications, such as distributed plant control systems, is more complicated and sophisticated in its implementation.

In this paper, we consider the problem of using ATM technologies in supporting hard real-time applications. We propose a real-time client-server programming model suitable for implementation in ATM networks. In order to guarantee the timing requirements, we introduce some software mechanisms such as the real-time channel handlers, bandwidth management controllers and resource reservation system. These software components are needed to ensure: i) sufficient ATM resources (buffer space, network bandwidth, etc.), ii) the use of only these resources which have previously been reserved. Thus, in our hard real-time applications we achieve the time constrained communications, end-to-end QoS guarantee for communication services (including video and audio) as well as the possibility of using the commercially available real-time operating systems.

Hard real-time communication in packet-switched networks has been studied, among others, in research papers [2, 14]. In the first paper of Ferrari, a scheme for real-time channel in the packet or cell switched networks was established. In [14] it was proved that in the point-to-point packet switched networks there is the ability to establish real-time channels.

Deterministic delay guarantees in the ATM networks were provided in the paper by S.-K. Kweon, K.G. Shin [8]. This approach employed the leaky bucket model as the input traffic description model. In the paper by Raha [12] new connection admission control (CAC) algorithm was presented that can support the hard real-time connections in ATM networks. This algorithm can realize traffic connection in the ATM networks. A new efficient ATM cell scheduling scheme for improving real-time ATM services was postulated in the paper [11]. The concept of real-time ATM-based protocol server was presented in the paper [1].

This paper is organized as follows. Section 2 presents the programming model of a hard real-time application in the ATM network. In section 3 we show that our application can be supported by the ATM technology. In section 4, we introduce new measures of performance evaluation of hard real-time applications in ATM networks. Finally, section 5 presents conclusions concerning future work.

2. Programming model of hard real-time application in ATM network

Our programming model of hard real-time application is based on real-time client-server concept. We assume that all physical components of networks (e.g. I/O devices, database storage system, sensors, etc.) are clients. In our approach a client is a subsystem which is automatically used by some system (for instance digital TV camera) or by some human operator. A server is a subsystem which controls or monitors all the others. All clients can request a server to send their own data and demand varying QoS parameters. All servers must be reliable and run continuously for a long time. A real-time client-server programming model is presented in Fig. 1.

In our approach we had divided all the programming components into two groups according to the following criterion: the server must have a programming addresses and the client does not. Therefore, we propose the master-slave concept of data exchange between them. However, we have admitted the service of exceptions.

In order to implement of each client component we propose new programming abstractions, i.e. real-time channel handlers and resource reservation controllers. These programming abstractions are different from the similar general purpose ones in that they are used in real-time operating systems. In particular, real-time channel handlers must determine the upper boundary of the end-to-end transmission delays. The second programming abstractions must predict all the needed resources (e.g. virtual channel, bandwidth, etc.) to utilize the network resources in an effective way.

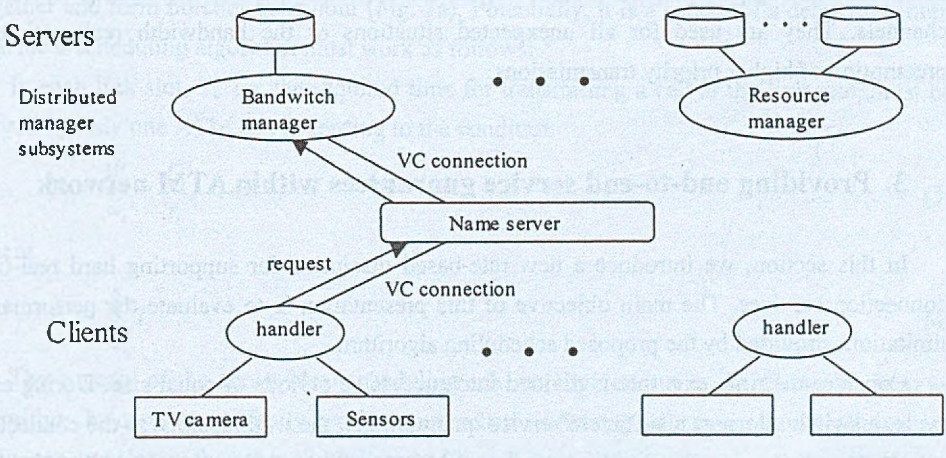


Fig. 1. Real-time client-server programming model suitable for implementation in ATM
Rys. 1. Model klient-serwer dla aplikacji czasu rzeczywistego w sieci ATM

In order to implement each server component we propose a new programming abstraction, namely the distributed manager subsystem. It is needed to monitor all clients, in particular, to acknowledge all client requests concerning its QoS requirements and the needed bandwidth for each link. Moreover, we anticipate the backup links that are needed in unexpected situations and in bandwidth reallocation to admit high priority transmission (see

Fig. 2.). Additionally, all programming abstractions allow to send time-constrained messages that must be guaranteed by the system.

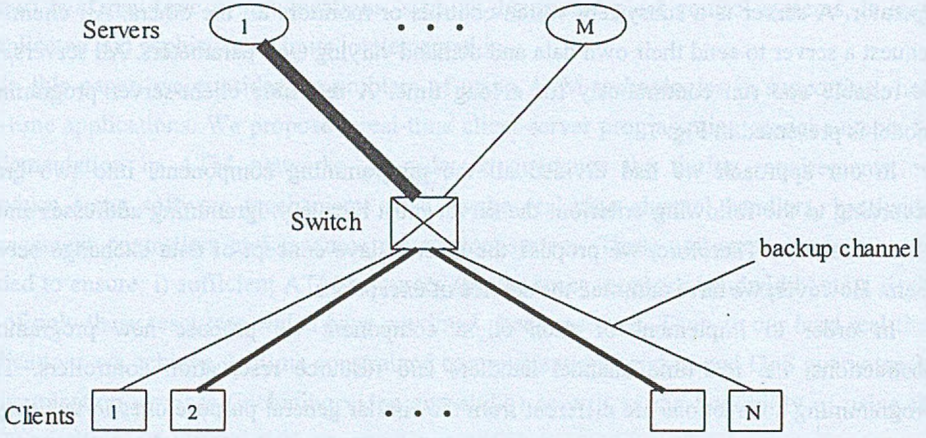


Fig. 2. Bandwidth allocation concept with backup channels
Rys. 2. Koncepcja alokacji pasma dla połączeń czasu rzeczywistego w sieci ATM

In the case where the server must cyclically serve all the clients, we need to divide the bandwidth among all of them. Moreover, in our approach we use the backup of virtual channels. They are used for all unexpected situations or the bandwidth reallocation in preemption of higher priority transmissions.

3. Providing end-to-end service guarantees within ATM network

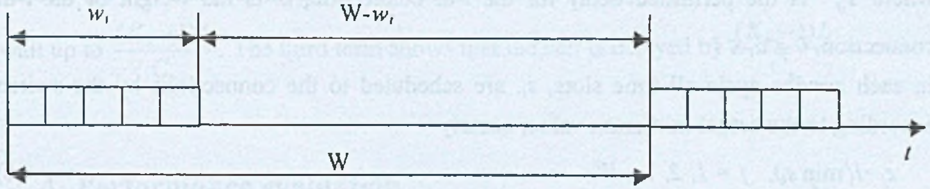
In this section, we introduce a new rate-based discipline for supporting hard real-time connection services. The main objective of this presentation is to evaluate the performance limitations provided by the proposed scheduling algorithm.

Consider the time axis that is divided into successive periods of equal size. During each cycle, a switch allocates a sufficient service quantum, i.e., the w_i time slots, to the connection i, $i \in K$. The service rate in each time slot is equal to

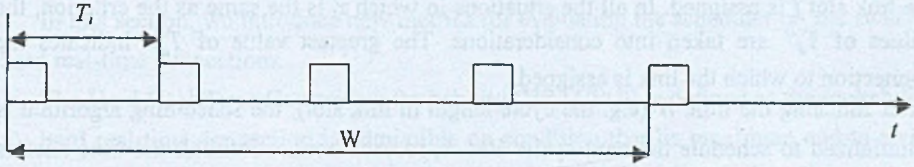
$$r_i = \frac{\mathrm{w}_i}{\mathrm{W}} C \tag{1}$$

where W denotes the cycle length (in link slots), C is the link capacity in [bps], K is the set of connections shared by the same link. It must be noted that it can be satisfied the following condition:

$$\sum_{i \in K} r_i \le C \tag{2}$$



a) Weighted Round-Robin Scheduling



b) Ideal Scheduling

Fig. 3. Output streams under Weighted Round-Robin Scheduling and Ideal Scheduling
Rys. 3. Strumienie komórek ATM jako wynik szeregowania z użyciem algorytmu WRR oraz tzw. idealnego schematu szeregującego

The point of the cell scheduling is given in Fig. 3. In one of the first scheduling algorithm, the weighted round-robin multiplexing [6], the output cells of a connection tend to cluster together and form burstier behaviour (Fig. 3a). Potentially, it is a source of a delay or a jitter. The ideal scheduling algorithm must work as follows:

In each link slot, T_i, i.e. the required time for transmitting a cell at the link rate, must be served by only one ATM cell according to the condition

$$T_i = \frac{W}{w_i} = \frac{C}{r_i}, \quad i \in K \tag{3}$$

where

$$W = \sum_{i=1}^{K} T_i \tag{4}$$

The concept of the above ideal scheduling scheme is given in Fig. 3b. However, such an algorithm belongs to perfectly worked out scheme. Among others, it is not allowed to influence one slot by the other and is assumed that T_i is an integer value.

Our scheme works as follows:

1. Each connection i has a priority, p_i, where $p_i = 0, 1, \ldots, K-1$, and $p_i \neq p_k$ for $i \neq k$. We said that connection i has a higher priority than connection k if $p_i < p_k$. Hence, the p_0 is the highest priority connection.
2. To the connection i, $i \in K$, is assigned the time slice that is equal to

$$s_j = p_i + a_i * T_d^{(i)}, \quad i \in K \tag{5}$$

where $T_d^{(i)}$ is the permitted delay for the i-th connection, a_i is the weight of the i-th connection, $0 \leq a_i \leq 1$.

3. In each service cycle all time slots, s_i, are scheduled to the connections by the switch according to the actual minimum value, namely

$$z_j = I(\min_{i \in K} s_i), \quad j = 1, 2, \ldots, W \tag{6}$$

where I is the identifier function $I(s_i) = i$, z_j is the identifier of the connection to which the link slot j is assigned. In all the situations in which z_j is the same as the criterion, the values of $T_d^{(i)}$ are taken into considerations. The greatest value of $T_d^{(i)}$ indicates the connection to which the link is assigned.

4. After finishing the time W (e.g. the cycle length in link slot), the scheduling algorithm is reinitialized to schedule the next cycle. Hence, to each connection link the time slice is assigned according to the equation (5).

Now, we evaluate the deterministic limit of the end-to-end delay suffered by a connection in the ATM system with the switches controlled by our scheduling algorithm. We assume after [9, 3, 7] that the source traffic of connection i is characterized by the (σ_i^0, ρ_i), where ρ_i is the drain rate and σ_i^0 is the bucket size in the leaky-bucket model.

We note that a source leaky-bucket is constrained by parameters (σ_i^0, ρ_i) if, at any interval of length t, it can transmit $\sigma + \rho t$ bits at most.

Let $g(i,h)$ be the service rate assigned to the connection i at the switch h, $1 \leq h \leq H$, where H is the number of the switches on the connection i. We suppose that $g(i)$ the smallest of the $g(i,h)$ on the connection i.

We assume that $g(i) \geq \rho_i$. Otherwise, the queue at one of the schedulers on the switch will build up without a bound.

The maximum end-to-end delay suffered by an H-hop connection i is bounded by [7]:

$$T_{d,\max}^{(i,H)} = \frac{\sigma_i^0 + H * L}{g(i)} + \sum_{h=1}^{H} \frac{(K_h - 1)L}{C_h} + \sum_{h=1}^{H-1} \frac{(K_h - 1)L}{C_h} \tag{7}$$

where

K_h - is the number of connections multiplexed into the outgoing link of switch h;

C_h - is the outgoing link capacity of switch h;

L - is the cell length (i.e. 53x8 bits).

Intuitively, the above result means that though the connection actually traverses a series of schedulers, it behaves as if it were served by a single scheduler with rate r_i, so that when the source sends a burst of length σ_i^0, it experiences a maximum delay $\frac{\sigma_i}{r_i}$. The second term reflects the fact that if a cell from connection *i* arrives at busy schedulers, the cell may have to wait up to $\frac{(K_h - 1)L}{C_h}$. The third term shows that the cell is delayed by $\frac{(K_h - 1)L}{C_h}$ at most.

4. Performance evaluation

In this section, we introduce new metrics for evaluating the scheduler on the switches for hard real-time connections.

1. The Hard Real-Time Connection Probability (HRTCP(*i*)) is defined as the probability that hard real-time connection is admissible on condition that he maximum end-to-end delay suffered by *H*-hop connection *i* is not exceeded, where *H* is the number of the switches on the connection *i*.
2. The Hard Real-Time Backup Probability (HRTBP(*i*)) is defined as the probability that hard real-time backup connection can be realized on the same condition as the above mentioned.

It is desirable to have a scheduler algorithm for the ATM network with the highest probability of HRTCP(*i*) and HRTBP(*i*). Our management layer may create the connection for each connection *i* with the allowable parameters of the maximum end-to-end delay, the HRTCP and the HRTBP.

Now, we evaluate the proposed scheduling algorithm implemented at all the switches along the connection. Therefore, we compare the maximum end-to-end delay, end-to-end jitter for the Packet Generalized Processor Sharing (PGPS) [10], the Self-Clock Fair Queueing [5], the Stop and Go (S&G) [4] and our scheme. For the purpose of comparison, we used the example presented in [13]. In the studied ATM network each VC connection is served by 5 switches. We suppose that our connection is characterized by $\sigma^0=50$ cells and $g=1$ Mbps. Each link is assumed to be 150 Mbps and shared by 200 connections. We have compared the performance of PGPS, SCFQ, S&G and our scheme in Table 1.

Table 1

Comparison results

Service discipline	End-to-end delay [ms]	End-to-end jitter [ms]
PGPS	23,3	23,3
SCFQ	26,1	26,1
S&G	107,0	107,0
Our scheme	22,75	22,75

5. Conclusions

In this paper, we have presented the details of the ATM supporting of hard real-time local area networks. The key issue in solving this problem is be obtain to required bounds on the end-to-end delays of the connections. Moreover, the use of backup connections is possible in order to avoid the unexpected situations such as preemptions by higher priority connection or the lack of a free bandwidth for a new call.

In this paper, we have introduced a new scheduling algorithm implemented at all the nodes along the hard real-time connections. Besides, we introduce a new performance metrics for evaluating hard real-time connections. It is obvious that these results draw near to introduction of the ATM technology into the distributed hard real-time systems. Hence, we expect that the next generation of the ATM LANs will be really realized.

REFERENCES

1. Borris M., Hartig H.: Design and Implementation of a Real-Time ATM Based Protocol Server. Proc. 19th IEEE Real-Time Systems Symposium, 1998, pp. 242 - 252.
2. Ferrari D., Verma D.: A Scheme for Real-Time Channel Establishment in Wide-Area Networks. IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, April 1990, pp. 368 - 379.
3. Figueira N., Pasquale J.: An Upper Bound on Delay for the Virtual Clock Service Discipline. IEEE/ACM Trans. on Networking, Vol. 3, No. 4, Aug. 1995, pp. 399 - 408.
4. Golestani S.: A Framing Strategy for Congestion Management. IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, Sep. 1991, pp. 1064 – 1077.
5. Golestani S.: Network Delay Analysis of a Class of Fair Queueing Algorithms. IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, Aug. 1995, pp. 1057 – 1070.
6. Katevenis M., Srdiropoulos S., Courcoubetis C.: Weighted Round-Robin Cell Multiplexing in a General Purpose ATM Switch Chip. IEEE Journal on Selected Areas in Communications, Vol. 9, Np. 8, Oct. 1991, pp. 1256 - 1279.
7. Keshav S.: An Engineering Approach to Computer Networking. Addison-Wesley, Reading, 1997.
8. Kweon S.-K., Shin K.G.: Providing Deterministic Delay Guarantees in ATM Networks. IEEE/ACM Trans. on Networking, Vol. 6, No. 6, Dec. 1998, pp. 838 - 850.

9. Parekh A., Gallager R.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. IEEE/ACM Trans. on Networking, Vol. 1, No. 3, June 1993, pp. 344 - 357.
10. Parekh A.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. IEEE/ACM Trans. on Networking, Vol. 2, No. 2, Apr. 1994, pp. 137 - 150.
11. Park Y.-K., Kim Y.-J.: An Efficient ATM Cell Scheduling Scheme for Improving Real-Time ATM Services. IEEE Trans. on Consumer Electronics, Vol. 47, No. 1, Feb. 2001, pp. 138 - 142.
12. Raha W., Zhao W., Kamat S., Jia W.: Admission Control for Hard Real-Time Connections in ATM LANs. IEE Proc.-Commun. Vol. 148, No. 4, Aug. 2001, pp. 217 - 228.
13. Zhang H.: Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. Proceedings of the IEEE, Vol. 83, No. 10, Oct. 1995, pp. 1374 - 1396.
14. Zheng Q, Shin K.G.: On the Ability of Establishing Real-Time Channels in Point-to-Point Packet-Switched Networks, IEEE Trans. on Communications, Vol. 42, No. 4, 1994, pp. 1095 - 1105.

Recenzent: Prof. dr hab. inż. Tadeusz Czachórski

Wpłynęło do Redakcji 8 kwietnia 2002 r.

Streszczenie

Zagadnienie zastosowania technologii ATM do budowy rozproszonych systemów nieprzekraczalnego czasu krytycznego jest nowym wyzwaniem. W pracy zaprezentowano model aplikacji typu klient-serwer, który może być zaimplementowany w sieci ATM. Zastosowano w nim nowe sterowniki, których zadaniem jest obsługa kanałów czasu rzeczywistego oraz rezerwacja zasobów. Pierwsze z nich wyznaczają górne ograniczenia na opóźnienia czasu transmisji, a drugie określają wszystkie zasoby (m.in. szerokość pasma transmisji), na które może być zapotrzebowanie w nieprzewidzianych sytuacjach (np. wywłaszczenie przez transmisję o wyższym priorytecie, uszkodzenie itp.). Dla zagwarantowania ograniczeń czasu krytycznego wprowadzono nowy algorytm szeregowania, który może być zaimplementowany w każdym przełączniku sieci ATM. Jego podstawowe parametry zostały porównane z innymi schematami (PGPS, SCFQ, S&Q). Do oceny połączeń

czasu krytycznego w takich systemach wprowadzono do literatury takie miary, jak HRTCP(*i*) – prawdopodobieństwo, że maksymalne opóźnienie typu „end-to-end" uzyskane przy przejściu przez maksymalną liczbę przełączników w *i*-tym połączeniu wirtualnym typu *VC* nie zostanie przekroczone, oraz HRTBP(*i*) – prawdopodobieństwo, że zapasowe połączenie może być zrealizowane na tych samych warunkach jak powyżej.