31

Fair and Flexible Contention Resolution for Input Buffered ATM Switches Based on LAN Medium Access Control Protocols

Andreas Kirstädter, TU München, Germany
andreas@lkn.e–technik.tu–muenchen.de

Abstract:

Only switches with input buffers offer the possibility to handle effectively the large and full–rate bursts that arise from the transport of data traffic and the burst accumulation within large ATM networks. Efficient contention resolution mechanisms are necessary to prevent output blocking in these input buffered switch architectures and to allow a fair and waste–free utilization of the switch.

This paper first reviews the different types of existing contention resolution mechanisms and shows their limits concerning fairness, scalability, and the possibility to support switching of multicast and prioritized cell streams. Then a new approach is presented that achieves an absolutely fair and efficient contention resolution on the cell level by using modified LAN medium access control (MAC) protocols. The requirements that a MAC protocol has to accomplish are investigated and the excellent performance (using an adapted version of the CRMA–II MAC protocol) of the proposed approach is shown. Finally native extensions toward the integration of multicast and prioritized traffic are given and it is demonstrated how this architecture can easily be scaled up to coordinate switches with throughputs of several Terabits per second.

Keywords

ATM Switch, Contention Resolution, Input Buffering, LAN, MAC Protocols

1 Introduction

During the last few years the application scenarios of ATM have changed dramatically. At the beginning the primary intention of ATM was the bandwidth–effective replacement of existing synchronous multiplexing hierarchies. By now it seems however that the biggest part of ATM traffic will be data traffic, i.e. ATM is going to be used increasingly for extending, replacing, and connecting the so called legacy LANs (802.x, FDDI).

The consequences of this development on switching architectures and their buffering strategies are no more neglectable: the switching and multiplexing of classic voice and video channels requires no big buffering efforts. So the switch concepts of the first years of the ATM life cycle focused on methods for reducing the connectivity expenses within the switch core (Ahmadi, H. and Denzel, W., 1989). And the performance analysts of that era used uncorrelated (i.e. non–bursty, bernoulli) traffic sources for exercising their models. But with the deployment of available bit rate (ABR) services for data traffic at large bandwidth delay products buffering strategies for ATM switches become increasingly important. Huge cell bursts will have to be managed by the switches during the reaction gaps of the data sources. And quite low cell loss guaranties have to be met in order not to disturb higher layer data communication protocols (ATM Forum, 1994). Corresponding simulation models have to use bursty traffic sources. At the same time future ATM switches will be confronted with the tasks of video distribution and handling different services classes so that multiple levels of priority and multicast traffic will have to be supported.

L. Mason et al. (eds.), *Broadband Communications*
© IFIP International Federation for Information Processing 1996

After reviewing buffering and buffer management strategies proposed in the literature shortly in the next section this paper presents a new buffer management scheme based on LAN medium access protocols that accomplishes all the requirements mentioned above. Two well known high speed LAN MAC protocols (CRMA–II and DQDB) are investigated concerning their suitability for the coordination of input buffered ATM switches. The resulting fairness is demonstrated and the delay throughput performance is compared to that of an ideal (output buffered) switch and a rate based control approach. Then a straightforward extension is shown that allows the coordination of an arbitrary and varying number of priority and multicast classes. Finally implementation, feasibility, and scalability aspects are considered.

2 ATM Buffering and Contention Resolution Strategies

ATM switches have to deal increasingly with highly bursty and asymmetric client–server data traffic and have to absorb large bursts during the reaction gaps of ABR controlled end systems (within networks with large bandwidth delay products). While it was shown that the best throughput delay performance can be achieved by pure output buffering (Hluchyj, M. G. and Karol, M. J., 1988) this strategy is simply not applicable to large numbers of switch ports and asymmetric load scenarios (Simcoe, R. J. and Pei, T.–B., 1995): in this case a relatively small speed–up of the output buffers (in the order of 4 to 8, s. e.g. Karol, M. J. et al., 1987) of the input bit rate, that was found to be sufficient in the case of symmetric load scenarios with bernoulli sources, no longer results in sufficiently low levels of cell loss so that the loss now becomes highly dependent on the applied load pattern. Shared memory buffering used by most current generation ATM switches (Garcia–Haro, J. and Jajszczyk, A., 1994) suffers not only from the fact that the necessary speed of the common buffer limits the maximum number of ports per switching module. The size of the common buffer is also severely limited since it has to be implemented on the (full custom) switch ASIC. Thus the only feasible buffering strategy is to use large input buffers that can easily be implemented at line speed.

Another problem arises from the need to coordinate the input buffers, i.e. to determine during each T_{cell} which of the contending input buffers is allowed to send a cell to a certain output. The main requirements for the contention resolution mechanism are:

- fair arbitration between contending inputs;
- waste–free operation (the max. switch throughput has to be impaired as less as possible);
- mechanisms for handling multicast and prioritized traffic;
- low implementation costs;
- independence from the special implementation of the switch core;
- scalability (i.e. a large maximum number of ports that can be coordinated).

Several contention resolution principles have been proposed in the past:

2.1 Usage of a Central Scheduler

Often the decision which input buffer may send a cell to which output port is taken by a central scheduling engine (s. figure 1, Obara, H., 1991, Matsunaga, H. and Uematsu, H., 1991). During

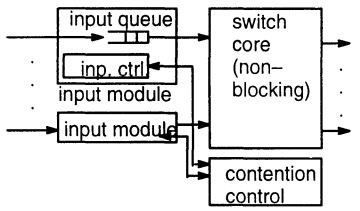

Figure 1: Centralized scheduling.

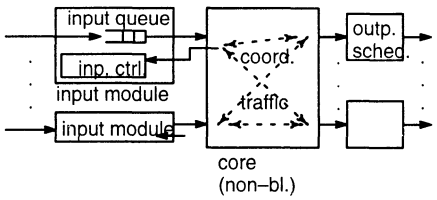

Figure 2: Output scheduling.

each T_{cell} it first receives information about all head of line (HOL) cells at all input buffers because otherwise its decision would be based on outdated information. Then it uses certain algorithms (e.g. round robin or neural network based etc.) to determine the winners between contending input buffers. This information is then transferred back to the input buffer control units that then at the end will send the corresponding cells into the switch kernel. Thus during the same T_{cell} a large amount of information has to be transferred and processed.

The scalability of this architecture is further limited if the input buffers are split into sub–buffers (one per switch output at each input, s. e.g. LaMaire R. O. and Serpanos, D. N., 1994) in order to prevent the HOL blocking effect present in simple FIFO input buffering solutions. Then the number of cells that have to be considered by the central scheduler rises with $o(N^2)$, N being the switch size. Also the usage of time stamps for circumventing the strict FIFO order increases the control data transfer and processing requirements (Obara, H., 1991).

2.2 Output Scheduling

A way to alleviate this problem by parallelism is the so called output scheduling (figure 2, s. e.g. Main, J. and Sarkies, K., 1995). Here a smaller decision engine is placed at each output port. All inputs having a cell ready for transmission to a certain output send a request to that output. The output then randomly selects one of the requesting inputs and notifies it by a confirm message. Some inputs may have got two or more confirm messages from different outputs. Since they can send only one cell during T_{cell} subsequent request–confirm rounds are necessary to allocate the excess confirm messages that those input returned to the outputs (a randomly chosen confirm is held back by each of the inputs).

So this parallel procedure substitutes the transmission and processing speed requirements of the central scheduling solutions above by a corresponding amount of hardware: slow but many ($o(N^2)$) lines are necessary for connecting each input buffer controller (IBC) with each output scheduler. Thus in a real systems an extra "switch" (maybe with its own contention resolution) is necessary for the transmission of the contention resolution information.

2.3 Dynamic Bandwidth Allocation (DBA)

The dynamic bandwidth allocation (DBA) (Worster, T. et al., 1995) tries to reduce the hardware expenses for contention resolution by allocating cell transmission rates instead of transmission instants for single cells to the inputs (s. figure 3). Each statistical multiplexing unit (SMU) at the input of the switch core sends a bandwidth request to the scheduler within the SMU at the corresponding destination output if one of several thresholds has been exceeded within any of its sub–buffers. The output SMU then considers this request together with the requests from other inputs and sends back an acknowledge signal indicating the cell rate that the requesting SMU can use. Request and acknowledge messages are transmitted by ATM cells through the switch core.

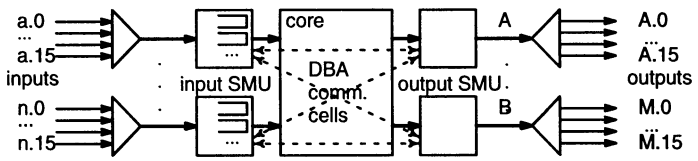

Figure 3: Architecture of DBA.

Since only cell rates are coordinated the switch core must offer high speed buffering capabilities (nearly as an output buffered core). Furthermore a low efficiency of this approach is to be expected since either a large amount of payload cells are used for the transport of coordination messages or the coordination itself will be rather slow and bandwidth wasting in the case of highly bursty traffic. These problems increase with the number of ports and if multicasts and priorities

are to be considered (the MUX / DEMUX solution for increasing the number of ports shown in figure 3 implies fairness problems at the output trunks A, B, etc.).

2.4 Two- / Three-Phase-Algorithms

Another family of coordination mechanisms for input buffers, back again on the cell level, are the so called 2- or 3-phase-algorithms (s. e.g. Hui, J. Y. and Arthurs, E., 1987). But since they can only be used together with a special kind of switch core (Banyan network with a bidirectionally transmitting Batcher sorter) they are not further considered in this paper.

2.5 Ring Reservation

A very promising approach for the coordination of input buffered ATM switches is the so called ring reservation (Bingham, B. and Bussey, H., 1988). A number of bit positions each controlling

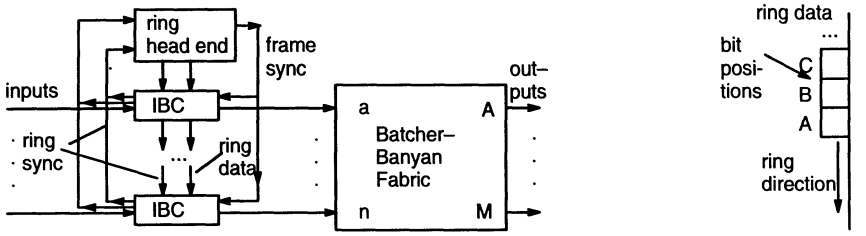

Figure 4: Ring reservation: architecture and ring data structure (control bits).

the sending of cells to a certain output (corresponding to the index of that bit) are clocked along a ring (a full rotation per T_{cell}) that connects all IBCs (s. figure 4). At the start of T_{cell} all bits have been set to "0" and an input can reserve for its HOL cell a sending opportunity for the next T_{cell} by successfully setting (from "0" to "1") the corresponding bit.

The big advantage of this solution is the strong reduction of the necessary coordination traffic and / or the corresponding hardware expenses since for the access control only a neighbor-neighbor communication in one direction between the IBCs has to be established. Splitting the input buffers internally into sub-buffers to avoid HOL blocking does not increase the necessary flow of coordination information. Neither a central controller is required nor are the core outputs involved in this communication. Further this mechanism is completely independent of the design of the core.

But so far the fairness problem arising from the sequential probing of inputs has not been solved successfully. The upstream one of two neighboring IBCs stays favored even if (as proposed in Bingham, B. and Bussey, H., 1988) the virtual origin of the bit frame is rotated (by a single IBC per T_{cell}) around the ring. Using a round robin mechanism by starting the search just behind the last serviced IBC for each output port is also not a solution to this problem. Not only the ring bandwidth would have to be doubled because now 2 rotations of the complete bit frame are required (since the sequence of the bit positions cannot be adapted) but also an unfairness in the case of coincidences (similar to that of DQDB, s. below) would occur. Furthermore up to now no satisfying native solution has been presented for the coordination of multicast and prioritized connections. Multicast cells have to be duplicated before entering the switch core (e.g. Xerox Corp., 1993) thus severely limiting the amount of multicast the switch can handle. Priorities can only be processed by rotating the bit frame a corresponding number of times per T_{cell} around the ring (each time coordinating a single priority level) thus inadequately increasing the ring bandwidth requirements.

3 Input Buffer Management Using LAN Medium Access Control Protocols

3.1 Basic Principles and Requirements

The idea behind this new solution for the buffer management problem is first to reduce the promising idea of ring reservation to its main principle: to tie the authorization for sending a cell to the successful setting of a certain bit position on a serial line. But instead of rotating the origin of the bit frame in a ring like fashion the proposed buffer management scheme uses always a fixed origin (making the ring to a bus and leading to new possibilities for an efficient handling of multicast and prioritized traffic). The fairness between contending inputs is guaranteed then by adapted medium access control protocols for bus networks from the local area networks domain.

Since the access of cells from the N switch inputs to each of the M outputs is controlled by its own medium access control this coordination system can be viewed as M virtual LANs (VLANs). All these VLANs exist on the same serial medium (the bit positions corresponding to the controlled outputs) and comprise N stations. Each input buffer controller (IBC) then comprises M finite state machines (FSMs) each of them controlling the transmission of cells of a single sub–buffer to its corresponding switch output.

The result is a LAN–switch hybrid system in the sense that the switched payload is still transmitted by input buffers into an ATM switch core but the coordination is done now by LANs whose bandwidth requirements have been reduced to a mere transmission of the fairness information.

Correspondingly the task of coordinating the cell stream from contending inputs can be considered to be split into three levels:

- The basic mechanism of preventing output blocking by the requirement to first set a bit before the corresponding cell can be transmitted.
- The superimposed medium access mechanism controlling the fair access of all inputs to all outputs.
- Intended interferences into the fairness mechanism to provide efficient means for the handling of different kinds of traffic (multicast, priorities).

The basic requirements the MAC protocol has to accomplish for each LAN in order to be suited for the coordination of input buffered ATM switches are:

- Low processing overhead: M parallel MAC FSMs have to be implemented within each IBC.
- Low consumption of bandwidth for the transmission of MAC fairness information between the IBCs: each VLAN can only use a few bits per T_{cell} on the serial line.
- Slotted transmission structure: the fairness mechanism itself has to be based on the assignment of throughput in discrete quantities (cells respectively bit positions).
- The MAC protocol must be able to work on bus LANs since they show the same asymmetry of access as the IBC control line.
- The MAC protocol must allow a full utilization of the LAN since no transmission capacity on a switch output is to be wasted (even if only one of the stations has traffic to transmit).

The above requirements can only be met by MAC protocols originally designed for high–speed bus LANs. During the investigation of several of those protocols for this task it has been found that another requirement arises from the fact that the M VLANs are not fully independent from each other. The underlying effect is that the hardware implementations of the switch core and the input buffers severely limit the number of cells an input buffer can send into the switch core during a single T_{cell}. So called coincidence situations arise when an IBC is able to occupy more bit positions within a single T_{cell} than the maximum number of cells it can emit during the same T_{cell}. The consequence is that the MAC protocol cannot rely on the fact that an IBC is able to use all the opportunities for setting bits it has been allocated to by the MAC protocol in order to restore its fairness.

3.2 Cyclic Reservation Multiple Access (CRMA–II)

Implemented on a dual bus topology a CRMA–II (van As, H. et al., 1991) network consists of a fairness scheduler (located in one of the two bus headends) and two counterdirectional buses. The buses are used for the payload transport between the stations and for conveying fairness information between the stations and the scheduler. Fairness is controlled in cycles that are initiated by a request frame emitted by the scheduler. The stations inform the scheduler about their fairness situation by entering into the request frame the actual number of slots pending within their transmission buffers and the number of slots transmitted since reception of the last request request frame. After the reception of the request frame (looped back by the other headend) the scheduler calculates a fairness threshold from the received values of the different buffer lengths and transmission counters (several algorithms have been proposed for this task in: van As, H. and Lemppenau, W. W., 1992, and Lemppenau, W. W. et al., 1993). The scheduler then emits a confirm frame containing the threshold value. At the reception of the confirm frame each station takes the appropriate measures to restore the fairness in a distributed manner: if its transmission count was above the threshold it defers a corresponding number of times its access to empty slots. Otherwise it is allowed to access a number of reserved slots issued by the scheduler. This number of confirms is the smaller value of either the difference between the threshold and the transmission count or the buffer length that it had previously informed the scheduler about. Before re–transmitting the confirm frame down the bus the station finally subtracts the threshold value from its transmission counter. Immediately following the emission of the confirm frame the scheduler generates a number of slots that are marked as reserved and can only be used by stations having calculated a positive number of confirms. So that the sum of marked slots equals the sum of confirms of all stations. After that the scheduler starts a new fairness cycle by emitting the next request frame.

In order to apply this cyclic fairness mechanism to the coordination of the IBCs the following "infrastructure" has to be provided between the IBCs (s. figure 5):

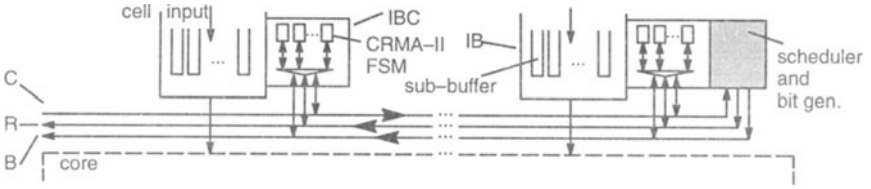

Figure 5: Architecture for the coordination of IBs by modif. CRMA–II.

- A scheduler entity generating the bit positions for all the VLANs is placed within the first IBC (before its FSMs) at the origin of the serial line B that conveys the bit positions that control the access to the outputs. For an increased reliability of the system the scheduler FSM could be present in each of the IBCs (only the first one being active).
- Each bit position on B has to be complemented by a second bit showing the reservation state of the bit position. This second bit may either be conveyed by a parallel line R (as shown in figure 5) or it could be time–multiplexed onto B.
- Finally a line C is necessary for the transmission of fairness information (transmission counters and buffer length values) from the IBCs to the scheduler. The threshold values for the different VLANs can be transmitted from the scheduler to the IBCs on the line B between the frames containing the bit positions.

  The duration T_{cycle} of the scheduling cycles of the single VLANs is mainly determined by the time T_t necessary for the transmission of the fairness information from the IBCs to the scheduler. At a switch size of 64x64 ports a T_t below 2* T_{cell} per VLAN has been found to

be sufficient leading to a T_{cycle} in the order of $128*T_{ell}$. If at much larger switch sizes a reduction of T_t is desired a few parallel lines can be used for C.

Also the CRMA–II protocol itself has to be adapted to the task of coordinating switch IBCs by the following modifications:

- The transmission counters are reset each cycle after notifying the scheduler about their values instead of decrementing them by the threshold value. Otherwise (depending on the difference between the loads of the single stations) the transmission counters would assume a large range of values (as in the original CRMA–II) and the resulting transmission time part (T_t) of T_{cycle} would lead to a slowed down reaction (determined by T_{cycle}) of the fairness mechanism.
- The deferment of stations with excess throughput is no longer used since the IBCs are only able to defer the access on those bit positions that have not been marked as reserved. A spatial re–use of bit positions (as with the destination release in the original LAN) does no longer exist so that during the marking phase each marked bit position also works as a implicit deferment (of the IBCs that cannot access it). The usage of deferments also becomes unreliable if not enough unmarked bit positions exist.
  Since now only confirms can be used for the equalization of throughput the scheduling threshold has to be always equal to the largest transmission counter value.
- At the beginning of the marking period each station pre–subtracts the number of calculated confirms from its transmission counter since they serve as the equalization of unfair throughput situations during the last cycle. Otherwise an IBC would be "punished" for the act of trying to get its fare share by accessing the reserved slots granted to him in this T_{cycle}.

3.3 Distributed Queue Dual Bus (DQDB, IEEE 802.6)

The distributed queue dual bus (s. e.g. Conti, M. and Lenzini, L., 1991) is usually the first MAC one comes to think of when looking for high–speed bus LANs. The fairness of the transmissions on one unidirectional bus is controlled by the emission of transmission requests in the opposite direction so that a global queue of data slots (not more than one per station) to be scheduled for transmission (by all stations) on a bus is constructed by implicitly considering the position of the stations.

Applying this fairness scheme to the control of the IBs in an ATM switch a topology similar to that of figure 5 (without the scheduler and line C) results where the line R is used this time to transfer the requests (each again a bit position) in the opposite direction. So for each of the M VLANs a distributed queue (DQ) is constructed for the transmission of cells to that output.

Complications arise in the case of coincidences when a IBC is in the HOL position of more than one DQ at the same time but can emit only one of the corresponding cells into the core. It has to pass the other sending opportunities to the downstream stations that were originally behind the losing station in the DQ: the DQ mechanism is severely disturbed resulting an unfair throughput behaviour. The number of places a station has to go back or advances within the DQ depends on the actual load pattern and the coincidence situation of the stations. Each single station cannot asses the actual throughput situation (i.e. its own throughput compared to that of every other station) on its own. Thus a distributed reconstitution of fairness (as in the case of the bandwidth manager in large DQDB networks) is not possible.

In the CRMA–II solution the scheduler compares each scheduling cycle the throughput values of the different IBCs on the different VLANs and is able to fully equalize throughput losses caused by coincidence situations within the regular scheduling. The result is an absolute fairness even at overload and asymmetric load situations (s. figures below).

4 Simulation Results

Each of the plots has been derived by time discrete process–oriented simulations at the cell level. Figures 6 and 7 compare the fairness behavior of DQDB and (modified) CRMA–II in the case

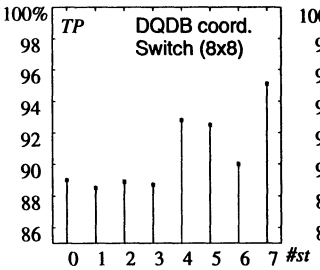

Figure 6

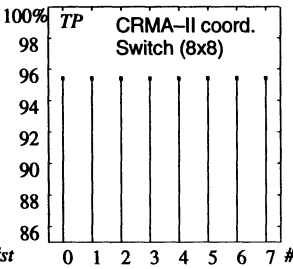

Figure 7

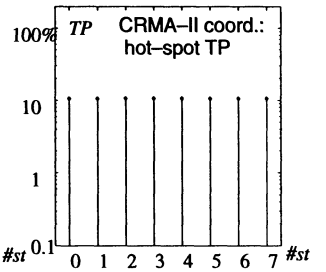

Figure 8

of a symmetric overload situation: the input links are fully utilized (normalized *load* = 100%) and the destination addresses of the incoming cells are equally distributed. Figure 7 also shows that the maximum achievable throughput of the CRMA–II controlled switch gets very close to 100%. In the figures 8 and 9 the same model of the CRMA–II controlled switch has been exer-

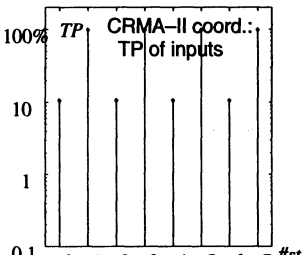

Figure 9

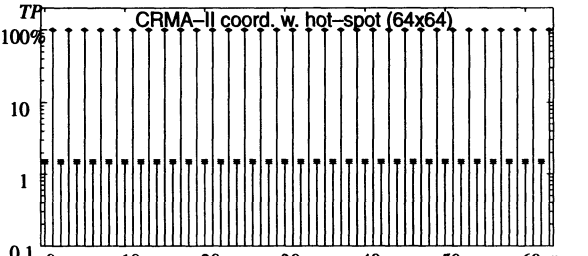

Figure 10

cised with extremely asymmetric traffic: while inputs with an odd index get the same load as before in figure 7, the cells on even inputs are directed only to the output with index 0 (the "hot–spot"). Figure 8 demonstrates the absolute fair sharing of this hot–spot output by the single inputs. Figure 9 shows the total throughput (as a sum) each input gets in this situation: the interesting fact is that the throughput values of the odd inputs are only neglectably impaired by the reactions of the fairness mechanism (the values get very close to 100%). In figure 10 a 64x64 switch under the same type of asymmetric load shows that this behavior is maintained even if much larger switches are considered.

The MAC coordination principle (for a 64x64 switch with a link bit rate of 150 Mbit/s) finally has been compared to two alternatives of the DBA method (discussed above) and an ideal (output buffered) switch of the same size. Two different types of cell sources have been used for the simulation of the CRMA–II coordinated switch: an uncorrelated type (where the destination of each cell is independent from the destination of previous cells) and an on–off type with full–rate bursts. In the latter case the mean burst size was 16 cells and the medium cell rate was adjusted by choosing the off–time accordingly. The data for the two DBA alternatives have been taken from (Worster, T. et al., 1995). The source model used in those simulations can also be considered to be uncorrelated since each of the inputs was loaded by a large number of independent small on–off sources (the peak bit rate being 10 Mbit/s at a medium bit rate of 0.1 Mbit/s and a mean burst size of 100 cells). Thus no significant increase of the probability that contiguous cells are destined

to the same output will be observed compared to the case of a random destination selection on a cell by cell base.

The comparison of the throughput delay behaviour in figure 11 shows several facts:
First the performance of the MAC coordinated switch under uncorrelated traffic gets very close to that of the ideal switch. Under full–rate bursts the performance of the new contention resolution mechanism is still comparable to the better one of the DBA architectures (having a core size of 4x4). The DBA architecture with the 600 Mbit/s SMUs (and a core size of 16x16) shows the worst performance. This might at least be partially attributed to the throughput degradation resulting from the much larger amount of coordination traffic required at larger core sizes.

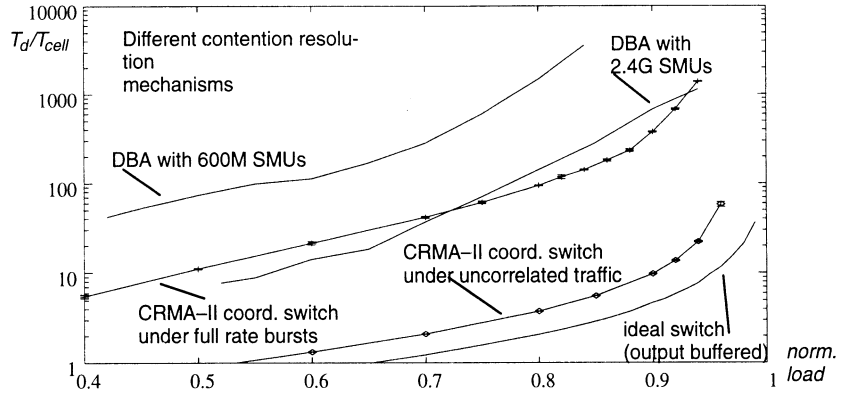

Figure 11

5 Integration of Multicast and Prioritized Cells

The above proposed MAC based control mechanism can be extended in a native way to permit differently prioritized inputs:

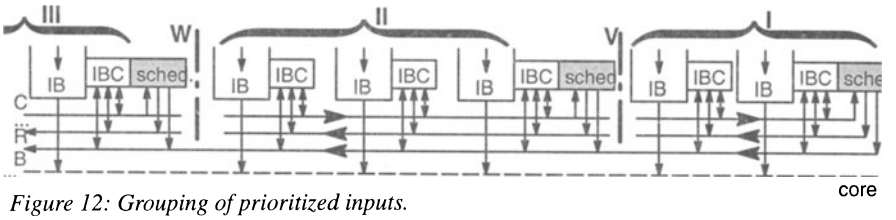

Figure 12: Grouping of prioritized inputs.

By splitting the lines R and C (at the positions W and V) and additionally activating the scheduler entities in the IBCs beneath each of those interruptions the IBCs are separated into several groups (s. figure 12). Since the R and C lines now only connect the IBCs within a single group fairness is only established between the members of this group. The line B that controls the access into the core still connects all of the IBC members of all groups. As the consequence the members of a group downstream on B can only access the bit positions left empty by the members of all upstream groups. Thus each group represents a own priority class of inputs.

This structuring principle can now be used to process correctly and with little overhead cells with an arbitrary number of priority levels that enter an ATM switch (s. figure 13).
The inputs of the switch are directly connected to the IBCs of the lowest priority group (LPIBCs). Cells of the lowest priority are completely managed by the IBCs of this group: they are stored

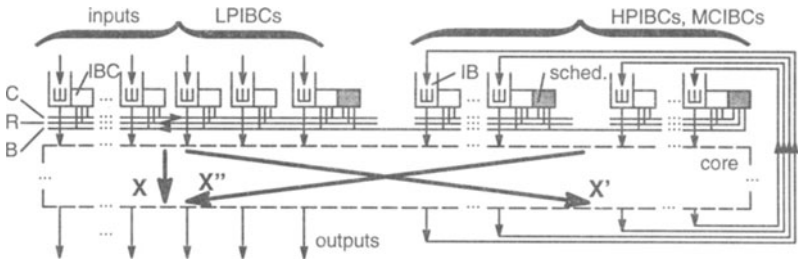

Figure 13: Flexible solution for multicast and prioritized traffic.

in sub–buffers corresponding to their destination output and are emitted into the core if the required bit positions are still found empty (path X in figure 13).

Cells of any higher priority level are processed by high–priority IBCs (HPIBCs) consisting of the members of the higher priority groups upstream. Corresponding to the expected throughput ϱ_i of cells of a certain priority class i a group of s_i IBCs is composed for the service of that class. A number of s_i specialized bit positions on the serial line control the access to the HPIBCs of that class. Any LPIBC manages a number of s_i special sub–buffers for each priority class i (i>0). A LPIBCs receiving a cell of a prioritized connection (priority level i) stores that cell in one of its s_i sub–buffers for that priority level. If the LPIBC receives a corresponding specialized bit position as empty it sets the bit on line B and is now able to transmit the cell to one of the HPIBCs of that priority class i (feed–back path X' in figure 13). The HPIBC then stores the cell in one of its sub–buffers and re–transmits it to its destination output (path X") if it receives the necessary empty bit position.

This server based principle exhibits a number of advantages compared to "traditional" solutions for the problem of coordinating multiple priorities:

- An arbitrary mixture of priority classes can be handled by the same infrastructure.
- The number of necessary HPIBCs always has an upper limit: since each of the N LPIBCs can receive and emit at most one cell per T_{cell} a maximum number of N HPIBCs (of all possible server groups together) is needed. The number of bit positions is also extended not more than by N. In the case of a symmetric switch the bit rate on the serial lines has only to be doubled to handle a nearly arbitrary number of priority classes. Traditional approaches always require an additional arbitration round per priority class.
- All the necessary HPIBCs will not need more space for their implementation than a single line card of the switch since each of them requires only a very small buffer capacity: in the case of an increasing buffer length a HPIBC can send an internal backpressure signal to all the LPIBCs at once by presetting the specialized bit position that controls the access of the LPIBCs to its own input.
- The HPIBCs can be flexibly re–partitioned into more appropriately sized groups in order to adapt the system to fluctuations of the relative throughput of the single priority classes. Only the lines C and R have to be split or re–connected at predetermined joints and some schedulers have to be activated respectively deactivated. Since just the assignment of sub–buffers to priority classes within the LPIBCs has to be adapted it is not necessary to interrupt the operation of the whole system. Only the access to the HPIBCs under reconfiguration has to be stopped for a few cell durations by presetting the corresponding specialized bit positions.

In the case of multicast (MC) connections this server principle also makes it unnecessary to have a cell duplication entity in each of the IBCs. One group of HPIBCs is assigned to the task of serving multicast cells: they become MCIBCs. Normally this will be the highest priority group but

also other solutions can be selected. The LPIBCs are informed about this fact by the configuration management and treat the cells of the MC connections using the same mechanisms as with the other priority classes: they are deflected into the corresponding sub–buffers and routed on the path X' through the core to the MCIBCs. Only there the duplication takes place as far as necessary (depending on the usage of different switch cores) and the copies are emitted into the core on path X" if the corresponding bit positions are available.

6 Implementation and Scalability

Most important for the scalability of the proposed approach is the number of switch ports that can be coordinated. Within one T_{cell} the frame with the bit positions (on the lines B and R) has to be completely clocked through all IBCs in order to control the emission of cells within the following T_{cell}. Assuming conservatively that the serial lines (and the serial parts of the IBCs) can be implemented with the same speed as the serial ports of the switch core a number of around 450 clock periods (length of an ATM cell plus internal header information) are available for this task. For the coordination of a symmetric switch (N input and output ports) with a medium number of HPIBCs and MCIBCs $1.5*N$ bits have to be shifted along $1.5*N$ IBCs. So in this case the maximum switch size will be N_{max}=450/3=150.

With a single serial line N_{max} is limited by the technology for connecting the serial lines to the IBC chips. A larger N_{max} therefore can be reached if parallel bundles are used. Each line within a bundle then conveys the bits belonging to a certain part of the switch outputs. The parallel lines within the bundle can mainly be processed in parallel. The only processing step that has to evaluate all of the lines of a bundle together is the reaction to coincidences: an IBC cannot set a number of bits (on all lines together) that exceeds the number of cells its input buffer can emit during T_{cell}. Thus the maximum number of coordinated ports is no longer limited by the interconnection technology used for the implementation of the serial lines. It is only limited by the maximum delay of a small number of gates within the IBC chip. So a much larger number for N_{max} is possible.

Another interesting point is the fact that not only a special family of switch cores (with its own very specific properties like speedup or buffering strategy) can be used. A great deal of the fundamental switch functions are now already done before the cells enter the core: e.g. the sorting of the cells corresponding to destination ports and the coordination of competing cells according to their priorities. Thus the switch core has only to deliver the pure connectivity function, i.e. to offer a way from any input to one or more outputs. This can also be implemented by using simple cross–bar or backplane solutions; with the cost–effective side effect that these core topologies offer the inherent capability of duplicating the cells within the core.

Summary and Future Activities

A new approach for the resolution of output port contentions in input buffered switches been presented that is based on the usage of MAC protocols of high–speed LANs. The requirements for the applied MAC protocol have been discussed and it has been shown that a modified version of CRMA–II provides a very high throughput and a complete fairness even under very asymmetric load scenarios. The proposed architecture can be easily extended to allow an arbitrary number of priority levels together with an effective coordination of multicast cells. Thus very large switches with throughputs of several Terabits per second can now be efficiently coordinated on the cell level.

The next step in the process of further examining this approach will be a VHDL–based hardware–level simulation of the critical parts of this topology (e.g. input buffer controller and scheduler) in order to further demonstrate its feasibility with common ASIC technologies. Also the investigation of two additional MAC protocols together with advanced simulations of the multicast and priority behavior are currently under way.

References:

Ahmadi, H. and Denzel, W. (1989): A Survey of Modern High–Performance Switching Techniques, IEEE Journal on Selected Areas in Communications, Vol. 7, No. 7, pp. 1091 – 1103

ATM Forum (1994): Reliability and Performance Considerations for ABR and VBR+ to support LAN Applications, Traffic Management Subworking Group, 94–0262

Bingham, B. and Bussey, H. (1988): Reservation–Based Contention Resolution Mechanism For Batcher–Banyan Packet Switches, Electronics Letters, 23 June 88, pp. 772 – 773

Conti, M. and Lenzini, L. (1991): A Methodological Approach to an Extensive Analysis of DQDB Performance and Fairness, IEEE J. on Sel. Areas in Comm. Vol 9 No 1, pp. 76 – 87

Garcia–Haro, J. and Jajszczyk, A. (1994): ATM Shared–Memory Switching Architectures, IEEE Network July / August 1994, pp. 18 – 26

Hluchyj, M. G. and Karol, M. J. (1988): Queueing in High–Performance Packet Switching, IEEE Journal on Selected Areas in Communications Vol. 6 No. 9, pp. 1587 – 1597

Hui, J. Y. and Arthurs, E. (1987): A Broadband Packet Switch for Integrated Transport, IEEE Journal on Selected Areas in Communications Vol. 5 No. 8, pp. 1264 – 1273

Karol, M. J. et al. (1987): Input Versus Output Queueing on a Space–Division Packet Switch, IEEE Transactions on Communications Vol. 35

LaMaire R. O. and Serpanos, D. N. (1994): Two–Dimensional Round–Robin Schedulers for Packet Switches with Multiple Input Queues, IEEE/ACM Trans. on Netw., pp. 471 – 482

Lemppenau, W. W. et al. (1993): ATM Implementation of the CRMA–II Dual Ring LAN and MAN, Proceedings of EFOC&N 93, The Hague, June 30 – July 2, 1993

Main, J. and Sarkies, K. (1995): Cell Scheduling Using Status Arrays in Input Buffered ATM Switches, Proceedings of the First IEEE Worksh. on Broadb. Switching Syst., Poznan, Polen

Matsunaga, H. and Uematsu, H. (1991): A 1.5 Gb/s 8x8 Cross–Connect Switch Using a Time Reservation Algorithm, IEEE Journal on Selected Areas in Communications, pp. 1308 – 1317

Obara, H. (1991): "Optimum Architecture for Input Queueing ATM Switches, Electronics Letters, 28th March 1991, pp. 555 – 557

Obara, H. and Hamazumi, Y. (1992): Parallel Contention Resolution Control for Input Queueing ATM Switches, Electronics Letters, 23rd April 1992, pp. 838 – 839

Simcoe, R. J. and Pei, T.–B. (1995): Perspectives on ATM Switch Architecture and the Influence of Traffic Pattern Assumptions on Switch Design, Computer Comm. Review, pp. 93 – 105

van As, H. et al. (1991): CRMA–II: A Gbit/s MAC Protocol for Ring and Bus Networks with Immediate Access Capability, Proceedings of EFOC/LAN 91, London, June 19–21, 1991

van As, H. and Lemppenau, W. W. (1992): Performance of CRMA–II: A Reservation–Based Fair Media Access Protocol for Gbit/s LANs and MANs, Proceedings of EFOC/LAN 92, Paris, June 22–24, 1992

Worster, T. et al. (1995): Buffering and Flow Control for Statistical Multiplexing in an ATM Switch, Proceeding of the ISS 95, Berlin, Germany, April 1995, Paper no. 488

Xerox Corp. (1993): A Switching Network, European Patent Application, Publication Number: 0 571 166 A2, 24 Nov 1993.