Credit-Flow-Controlled ATM for MP Interconnection: the ATLAS I Single-Chip ATM Switch

Manolis Katevenis†, Dimitrios Serpanos†, Emmanuel Spyridakis†

Institute of Computer Science (ICS), Foundation for Research and Technology – Hellas (FORTH)
Science and Technology Park of Crete, Vassilika Vouton, P.O.Box 1385, Heraklion, Crete, GR 711 10 Greece
http://www.ics.forth.gr/arch-vlsi E-mail: katevenis@ics.forth.gr Tel.: +30 (81) 391664, fax: 391661

Proc. 4th Int. Symp. on High-Perf. Computer Architecture (HPCA-4), Las Vegas, Nevada, USA, February 1998
`ftp://ftp.ics.forth.gr/tech-reports/1998/1998.HPCA.atlas4mp.ps.gz`

Abstract

Multiprocessing (MP) on networks of workstations (NOW) is a high-performance computing architecture of growing importance. In traditional MP's, wormhole routing interconnection networks use fixed-size flits and backpressure. In NOW's, ATM –one of the major contending interconnection technologies– uses fixed-size cells, while backpressure can be added to it. We argue that ATM with backpressure has interesting similarities with wormhole routing. We are implementing ATLAS I, a single-chip gigabit ATM switch, which includes credit flow control (backpressure), according to a protocol resembling Quantum Flow Control (QFC). We show by simulation that this protocol performs better than the traditional multi-lane wormhole protocol: high throughput and low latency are provided with less buffer space. Also, ATLAS I demonstrates little sensitivity to bursty traffic, and, unlike wormhole, it is fair in terms of latency in hot-spot configurations. We use detailed switch models, operating at clock-cycle granularity.

† also with the Department of Computer Science, University of Crete, Greece.

1. Introduction

Multiprocessing (MP) on networks of workstations (NOW) is a high-performance computing architecture of growing importance [ACPN95]. In the evolution from traditional MP's to NOW's, interconnection networks resemble or become high-speed local area networks (LAN's). *Wormhole Routing* [DaSe87] is the most popular network architecture in traditional multiprocessors. With regard to local area networking, *Asynchronous Transfer Mode (ATM)* [LeBo92] is one of the major contending technologies.

There are notable analogies between ATM and wormhole routing. In an ATM network, variable size packets are fragmented into fixed size *cells* before entering the network. Cells are transmitted over virtual circuits (VCs), which are fixed routes over the network. All cells of a packet are assigned the same VC and are transmitted in order. In a wormhole network, variable size packets are fragmented into fixed size *flits* before entering the network. Flits of the same packet are routed in-order, following the same path from source to destination. Thus, ATM cells are analogous to wormhole flits. The flit size of wormhole routers has increased over the years, and is likely to become comparable to the size of ATM cells in the future.

Wormhole routing uses credit-based flow control (backpressure). In ATM, credit-based flow control can be added to the network, e.g. as the Quantum Flow Control (QFC) Alliance does [QFC95]. We are implementing *ATLAS I,* a single-chip ATM switch with credit-based flow control. This 4-million-transistor 0.35-micron CMOS chip will offer 20 Gbit/s aggregate I/O throughput, submicrosecond cut-through latency, 256-cell shared buffer containing multiple logical output queues, priorities, multicasting, and load monitoring.

This paper compares the credit flow control protocol of ATLAS I –which is similar to QFC– to the corresponding protocol of multi-lane wormhole routing [Dally92]. In the ATLAS I and QFC protocols, each cell must acquire two kinds of credits before it can depart: a per-connection (per destination) credit, and a per-link credit. Similarly, in the multi-lane wormhole protocol, a flit must acquire two resources before it can depart: a lane, and a credit inside its lane. However, the wormhole protocol differs in two important respects from the ATLAS I protocol (and from QFC), as analyzed in this paper. First, wormhole routing dedicates each lane to a single packet, for the entire duration of that packet's transmission; in contrast, ATM allows a lane (a buffer slot) to be used by any VC after the departure of the cell (not packet) that previously occupied it. Second, wormhole routing allows packets that go to the same destination to occupy multiple lanes per link at the same time; in ATLAS I, on the other hand, lane usage is governed by destination (flow group): identically-destined cells are not allowed to occupy more than one lane.

This paper shows that, when using the ATLAS I credit flow control protocol, a small buffer size is enough in the case of uniform traffic, and a few lanes suffice in the presence of bursts, while wormhole requires significantly larger buffer space and more lanes for similar performance, especially when large packets (bursts) exist. In the case of hot-spot traffic, a number of ATLAS I lanes which is slightly larger than the number of hot spots is enough for the non-hot-spot traffic to stay virtually unaffected by the presence of hot spots; in wormhole, even a quite larger number of lanes cannot offer satisfactory isolation.

The performance of wormhole routing was studied in [Dally92], but that study was restricted to a single packet size and to uniformly-destined traffic. For ATM, efficient credit-based flow control schemes have been studied for long-distance links [KuBC94] [OzSV95].

This paper differs from the above studies in the following points. For wormhole routing, we extend the study of [Dally92] by simulating the effect of various packet sizes and of hot-spots. For ATM, we study credit flow control with dynamic buffer sharing for *short links,* like those typically found in LANs and multiprocessor networks, rather than the long links of WANs studied elsewhere; in our case, the throughput-round-trip-time product of the links is less than one cell time. Also, our study of hot-spot traffic, of burst-size dependence, and our network topology and size differentiate ours from previous studies. In addition, we perform the same simulations for wormhole and ATM, and we compare and contrast the results. Our simulation models are detailed and realistic; they operate at *clock cycle* granularity, and they are based on the ATLAS I chip organization.

Section 2 presents an overview of the ATLAS I single-chip ATM switch. Section 3 describes the ATM credit-based flow control protocol studied in this paper. Section 4 points out the similarities and the differences between credit-flow-controlled ATM and wormhole routing. Section 5 presents the assumptions and models used in our simulations. Finally, section 6 presents and interprets the results of the simulation experiments.

2. Overview of the ATLAS I switch chip

ATLAS I (ATm multiLAne Switch I) is a general-purpose, single-chip gigabit ATM switch, being developed within the "ASICCOM" project of the European Union *ACTS* Programme. It is intended for use in high-throughput and low-latency systems ranging from wide area (WAN) to metropolitan (MAN), local (LAN), and system/desktop (SAN/DAN) area networking. It is appropriate for supporting a mixture of services ranging from real-time, guaranteed-quality-of-service to best-effort, bursty and flooding traffic. These networks can provide a range of services from telecom to multimedia and multiprocessor NOW (network of workstations) applications.

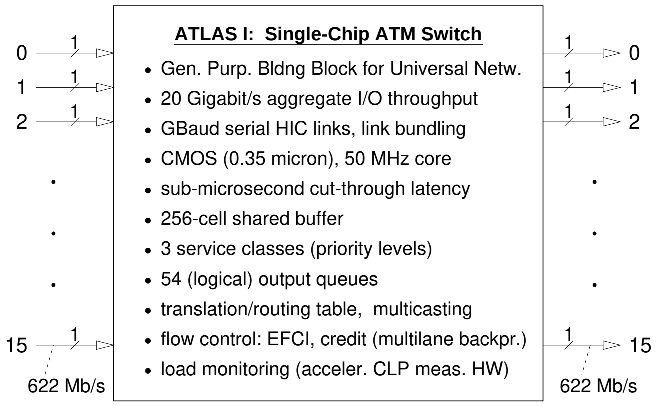

Figure 1: ATLAS I chip overview

Figure 1 presents an overview of this 4-million-transistor chip. It is a 16×16 ATM switch, with serial links of 622 Mbits/s each. Using link bundling, ATLAS I can also be configured as 8×8 at 1.25 Gb/s, or 4×4 at 2.5 Gb/s, etc. The ATLAS I links run ATM on top of IEEE Std. 1355 "HIC/HS" physical layer, using BULL's "STRINGS" GBaud serial-link transceiver. HIC/HS was preferred over SDH/SONET because of simpler circuitry, lower latency, and the capability to encode (unbundled) credits. Internally, the chip contains a 17×17 switch; the 17th link serves the chip control and monitoring circuits and the optional external management microprocessor. The ATLAS I chip operates as a crossbar with a 256-cell shared buffer; 51 logical output queues and 3 multicast queues are maintained in this shared buffer. ATLAS I implements three levels of priority, each level having its own queues; the output scheduler empties higher-priority queues before considering lower-priority ones. The chip contains a routing and VP/VC translation table; each table entry contains a bit mask, specifying the link or set of links over which the cells are to be unicast or multicast. The crossbar and shared buffer implement virtual cut-through; owing to this and to the all-hardware control, cell latency through the switch chip at low network load is much shorter than one microsecond. The control section of ATLAS I is shared among all incoming and outgoing links, and consists of two 5-stage pipelines. One pipeline handles cell arrivals and departures, while the second processes credit arrivals; each pipeline operates at the rate of one event per clock cycle (20 ns). More details on the architecture and internal organization of the chip can be found in [KaSV96].

At the time of this writing, ATLAS I has been fully designed, and placement and routing is in progress. The chip contains approximately half a million transistors in logic, and half a million bits of SRAM; the majority of the chip uses semi-custom logic, with a small full-custom part [Korn97]. ATLAS I will be fabricated by SGS Thomson, Crolles, France, in a 0.35 micron CMOS process, inside a 225 mm^2 die.

3. Credit-based flow control

Under credit-based (backpressure) flow control, network traffic is never dropped, because it is only transmitted when buffer space is known to exist at the receiver: when the transmitter has *credit* for such buffer space. The buffer space that is necessary and sufficient is equal to the *peak throughput* times the *round-trip time* (RTT) of data and credits. Single-lane backpressure indiscriminately starts or stops the transmission of any flit or cell according to buffer space availability. This causes poor performance, due to head-of-line blocking effects.

Multi-lane backpressure solves this performance problem by selectively allowing unblocked traffic to proceed, while preventing congested connections from occupying all available buffer space thus delaying all other traffic as well. In multi-lane wormhole routing [Dally92], a number L of lanes exist on a link. When a new packet arrives at the link entrance, it waits until a free lane exists, and then it acquires such a lane. Flits of packets that have acquired a lane can be transmitted, in any interleaving fashion, on the link; each flit carries the lane identifier of the packet to which it belongs. The receiver buffer is partitioned into separate space for each lane; each credit is for one particular lane; a flit can only be transmitted if it acquires a credit for its lane. After the last flit of a packet is transmitted, its lane is freed. Multi-lane wormhole routing has been implemented in hardware (e.g. [DaSe87], [Bork90], [CaVe96], [Gall97]).

In ATM, the simplest way to introduce VC credit flow control (multi-lane backpressure) is to dedicate separate buffer space to each connection (VC), and hence to use a separate credit counter for each VC; the buffer space per VC must be equal to the VC's peak throughput times the round-trip-time. This *static* buffer allocation ensures complete independence of each VC from all others, but it is also quite wasteful in terms of memory space. The required buffer space can be significantly reduced by *dynamically sharing* the receiver's buffer among connections. Such dynamic buffer allocation schemes have been developed for wide area networks [OzSV95] [KuBC94], where the amount of memory needed and the expected savings justify the use of protocols with some complexity. QFC [QFC95] is another such protocol, being standardized through the ITU-T; commercial products implementing QFC have been announced. Figure 2 illustrates a credit-based flow control protocol that is similar to QFC, but is adapted to hardware implementation over short and reliable links. This is the ATM protocol being simulated in this paper; ATLAS I implements the special case of this protocol where b = 1.

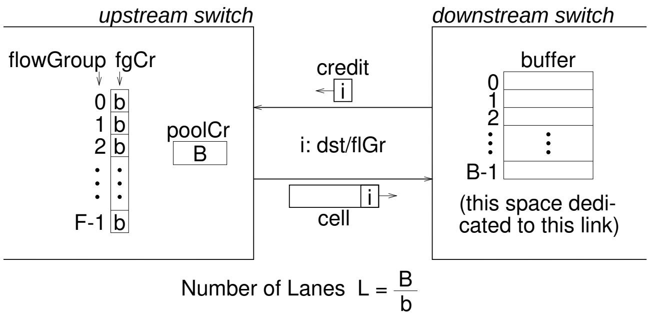

Figure 2: QFC-like ATM credit protocol used in simulations (ATLAS I uses b=1, hence L=B)

In ATLAS I, credits operate on the granularity of *flow groups,* because it may be undesirable or infeasible for them to operate on individual connections when the number of connections is large. A flow group is a set of connections over a common path; their cells never overtake each other. ATLAS I links can carry up to 4096 flow groups, each; the chip can *merge* multiple incoming flow groups into an outgoing flow group. In our simulations, all VCs with a common destination are grouped into a single flow group, regardless of origin.

For each link, the downstream switch allocates a buffer pool of size B. The upstream switch maintains a *pool credit* count per output, corresponding to the available space in that pool. This space is shared among cells according to their flow group. Separate credit counts, $fgCr$, are maintained for each flow group, initialized to b. The pool credit corresponds to QFC's $Limit_{link}$, and $fgCr[i]$ corresponds to QFC's per-connection $Limit[i]$. A cell belonging to flow group i can depart if and only if $fgCr[i] > 0$ and $poolCr > 0$; $fgCr[i]$ and $poolCr$ are then decremented by 1, each. When that cell leaves the down-stream switch, a credit carrying its flow group ID, i, is sent up-stream, where it increments $fgCr[i]$ and $poolCr$. Since the buffer pool contains B cell slots and each flow group is allowed to occupy at most b of these slots, at least $L = B/b$ of the flow groups can share this buffer pool at any given time; thus, L is the *number of lanes.*

ATLAS I uses the protocol of figure 2 with $b = 1$, so as to maximize the number of lanes $L = B/b$; also, the hardware implementation is simplified in this way. In ATLAS I, each credit carries permission for a single cell, and is transmitted upstream as soon as the receiver forwards one cell, unlike QFC where multiple credits are bundled together for transmission. ATLAS I is fast enough for the cell-credit round-trip-time to be less than one cell-time, for links up to a few meters in length; thus, even a single flow group is able to saturate a link by itself. In case more than L flow groups are ready for transmission on a link, those in excess of L are held by lack of pool credits; in that case, ATLAS I assigns newly arriving (pool) credits to waiting flow groups in FIFO order, which amounts to round-robin service, since each flow group returns at the back of the queue after being serviced once; round-robin is also used when allocating flow-group-credits to multiple incoming flow groups that are being merged into a single outgoing flow group.

4. Comparing ATM versus Wormhole

Cells and Flits: as discussed in section 1, there are notable analogies between ATM and wormhole routing, the most important of which is the use of fixed-size cells or flits for the transmission of packets. The cell size in ATM is 48 bytes, plus a 5-byte header. The flit size in wormhole routers has increased over the years: from 1 byte originally, it went to 4 bytes e.g. in iWarp [Bork90]; today, one finds wormhole routers with flit size equal to 8 bytes (the Intel Cavallino [CaVe96]), or 16 bytes (the SGI Spider [Gall97]), or 32 bytes (the ST C104 [Inmo93] and RCube [Zerr96]). The reasons for this increase are: *(i)* the increasing size of the smallest quantities that one usually ships over a network; *(ii)* the increasing throughput-RTT (round-trip-time) product of network links; *(iii)* the desire to keep controller complexity manageable, i.e. the desire for the flit rate to grow slower than the bit rate; and *(iv)* the desire to reduce the overhead of the flit header relative to the payload. With this tendency, cells and flits will soon have comparable sizes. To factor out this parameter from our study of flow control protocols, we assumed equal cell and flit payload sizes in our simulations.

Connections: ATM is connection-oriented, while wormhole is connection-less. Still, in wormhole, the equivalent of a "connection" is created per packet: the head flit stores route and lane assignment state, for use by subsequent flits of the packet. ATM is criticized for requiring a lengthy connection set-up phase; however, if "permanent" connections are set-up ahead of time, as is possible in networks with up to many thousand nodes, this connection overhead is taken out of the critical path. The existence of connections in ATM, allows the segregation of cells into flow groups and the efficient use of lanes by flow groups, as explained below. Achieving a similar segregation in wormhole is quite difficult, and has rarely been done.

Hardware Complexity differs between ATM switches with backpressure and wormhole routers, but in our view it does not differ dramatically [KaSS96]. The number and nature of the control memories is comparable in the two architectures, although their size is larger in ATM. In general, ATM costs more in the VP/VC translation table, and in the per-flow-group credit counts; wormhole, on the other hand, has the additional task of managing packets that wait for lane availability on the desired output link(s).

Use of Lanes: although the hardware complexity is comparable for ATM and wormhole switches with equal buffer size, their different backpressure protocols yield very different performance, as will be seen in section 6. Thus, ATM needs quite less buffer memory than wormhole to achieve comparable performance, while in some cases (hot-spot traffic) ATM performs better regardless of how much more buffer memory wormhole may have. This difference in performance is due to the different use of lanes in the two architectures, as explained below.

Lane Sharing: in networks with one or few lanes, the most important disadvantage of wormhole routing is that it dedicates each lane to a *single packet,* and does so for the *entire duration* of that packet's transmission. When the packet's progress is blocked because of contention, buffer resources are tied down. If the entire packet does not fit in one buffer, its tail blocks the lane that it has acquired on one or more links; when few lanes exist, all of them can get blocked in this way, causing the link to stay idle. By contrast, ATM allows a lane to be used by any connection after the departure of the cell (not packet) that previously occupied it. An ATM lane can even be shared among multiple connections at the same time, when lanes have more than one buffer slots each ($b > 1$ in figure 2)[1].

[1] this bears some analogy to [TaFr88], although [TaFr88] places a rather liberal upper limit on the portion of the buffer that each queue is allowed to occupy.

Lane Hogging: in networks with many lanes, the most important negative characteristic of traditional wormhole is that it allows packets that go to the same destination to occupy multiple lanes per link at the same time. Such similarly-destined packets may originate from the same or from different sources. No advantage is gained by doing this, since all these packets will get blocked if one of them does, while, on the other hand, all these lanes remain unavailable for packets traveling to other (less congested) destinations. In ATLAS I, lane usage is governed by destination (flow group): identically-destined cells are not allowed to occupy more than one lane, regardless of where they come from and which packet they belong to.

Dynamic (Adaptive) Routing is one aspect of wormhole that is *not* studied in this paper. Under dynamic routing, if the network topology includes multiple paths from a given source to a given destination, then different packets may follow such different paths (at the risk of out-of-order delivery). We could not study this property of wormhole because ATM does not allow it, since ATM guarantees in-order delivery. Adaptive routing can improve the performance of wormhole networks in some cases. However, the inefficient use of lanes by wormhole that was discussed above applies to the case of adaptive routing as well: lanes are still dedicated to single packets at a time, and multiple lanes, *in multiple paths* now, can be occupied by slow-moving hot-spot packets, thus becoming unavailable for packets going to uncongested destinations.

5. Simulation models

Two models were developed for the ATM and wormhole switches, in order to compare them through simulation. The models are quite realistic and detailed, operating at the granularity of individual clock cycles. The ATM model is based on the ATLAS I chip, although there are some differences between the two. The wormhole switch model was developed based on the ATM model, introducing only the necessary changes. With this approach, all non-essential characteristics (e.g. data buffer cell/flit staggering, credit exchange, etc) are made the same in the two models, and the two back-pressure protocols are evaluated under comparable circumstances. Below, we summarize the model features that are needed in order to understand the results of the simulations; the full details of the models can be found in [KaSS96].

- *Time Units.* All results are reported in units of *cell time.* This is the time taken to inject an ATM cell on a link. Time in the simulation flows in finer units: *clock cycles.* In our simulations, the cell time was defined as *fourteen (14)* clock cycles, reflecting an assumed processing of cells at the rate of 4 bytes per cycle (53/4 = 13.25).
- *Flit Size.* The payload of a wormhole flit is assumed to be the same as the payload of an ATM cell; so, a packet is fragmented into the same number of cells or flits. The header of wormhole flits is smaller than the ATM cell header; to account for this, we assume that the time taken to inject a flit on a link is *thirteen (13)* clock cycles. This also accounts for the wormhole credits being shorter than the ATM credits.
- *Link Capacity.* All network links, including source and destination, have a capacity of 1 cell per cell time in ATM, or 14/13 (=1.077) flits per cell time in wormhole.
- *Link Latency.* All links of the simulated networks, including source and destination links, have a propagation delay of 0.07 cell times (1 clock cycle), thus simulating a multiprocessor interconnection network where all processors are in close physical proximity.
- *Cut-Through Delay.* The switches modeled feature cut-through capability, like ATLAS I. The cut-through delay of the switch is 0.3 cell times (4 clock cycles) for ATM and 0.36 cell times (5 clock cycles) for wormhole; the additional clock cycle in wormhole accounts for lane assignment. The cell-credit round-trip time is 0.57 cell times in ATM, and 0.64 cell times in wormhole.
- *Network Topology and Size.* We simulated banyan (butterfly) networks; figure 3 shows such a banyan topology for a small size network. All results reported in section 6 are for a 6-stage network made of 2×2 switches, i.e. a 64×64 network. We also studied other network and switch sizes; see [KaSS96] for those results.

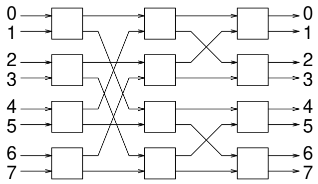

Figure 3: A 3-stage 8×8 banyan network

- *Connection Set-Up.* In ATM, we assume that "permanent" connections have been set up, ahead of time, from each source to each destination; no set-up overhead is included in our simulations. On each ATM link, all connections going to a common destination are grouped into a single flow group, regardless of their origin.
- *Traffic Sources.* There is one producer of cells/flits for every incoming link to the network. Producers produce fixed sized packets, each composed of a fixed number of cells/flits, with exponential interarrival time; in one set of experiments, sources produce a mix of small and large packets. Thus, the input is Poisson at the packet level, but bursty at the cell/flit level, capturing important characteristics of realistic network traffic. The Poisson model is

used as a standard reference traffic pattern, because our objective is a qualitative comparison of the architectures; the results show that the difference in performance is so large that the precise traffic model becomes less important. Fixed size packets were chosen for comparison of our results to those of [Dally92]. Each generated packet is targeted to a destination that is chosen at random. In all but the hot-spot experiments, packet destinations are chosen uniformly.

- *Input Load.* This is the average rate of packet generation in each traffic source, measured in cells or flits per cell time (i.e. same units as the link capacity).
- *Source Queue Organization.* Each producer has one separate queue per destination. In the ATM network, cells (destinations) that acquire credits depart from these queues in chronological order of generation. In the wormhole network, packets (of any destination, i.e. from any queue) acquire available lanes in chronological order of generation, with the added restriction that packets from a given queue (going to a given destination) are not allowed to acquire more than one lane at a time. This latter restriction is favorable to the wormhole architecture but is not a standard part of it; it was added because, in the hot-spot simulations, without it, the entire network was flooded by hot-spot packets, leading to catastrophic performance.
- *Shared Buffering.* The simulated switches use shared buffering, like ATLAS I. As shown in section 6.1, this performs better than the input queueing (single-lane) or input buffering (multi-lane) organizations that were simulated in [Dally92], while implementation cost is similar for the two organizations [KaVE95]. Note that the shared buffering organization that we simulate concerns the elimination of head-of-line blocking inside each lane; from the point of view of space allocation, static partitioning per incoming link is used (otherwise, credit management becomes too complex).
- B, b, L: as defined in figure 2 (B, b in cell or flit units).
- *Saturation Throughput.* One measure of performance that we report, for uniformly destined traffic, is the average outgoing link throughput when the input load exceeds 1; we measured this under input load equal to 2.0. We measure throughput in cells or flits per cell time (same units as input load and link capacity).
- *Delay.* Delay is our other measure of performance. We report total cell/flit delay, i.e. the sum of the queueing delay in the traffic source plus the network delay, for individual cells or flits –not for entire packets.

6. Measurements and interpretation

The models have been coded to obtain a simulator. We used CSIM [CSIM90], with C as the implementation language of the simulator. We report our results, beginning with uniformly destined traffic and fixed packet size. Then, we study the effect that the packet (burst) size has. After that, we add hot-spot traffic to the network, and we examine how that affects the non-hot-spot packets.

6.1 Uniformly-destined fixed-size packets

In these experiments on the 64×64 network, packets have uniformly distributed destinations, and the packet size is fixed at 20 cells or flits, i.e. the same as in [Dally92]; section 6.2 looks at other packet sizes. Figure 4 plots saturation throughput. ATM performs better than wormhole for a given buffer size and number of lanes, for the reasons explained in section 4. The ATM throughput approaches the link capacity with buffer sizes as small as 4 or 8 cells; by contrast, wormhole requires 16-flit buffers or larger in order to provide similar throughput. Another observation is that the ATM performance is good almost independent of the number of lanes (owing to lane sharing), while wormhole needs multiple lanes in order to achieve good saturation throughput; as will be seen later, lanes are needed in ATM when bursty or hot-spot traffic exists.

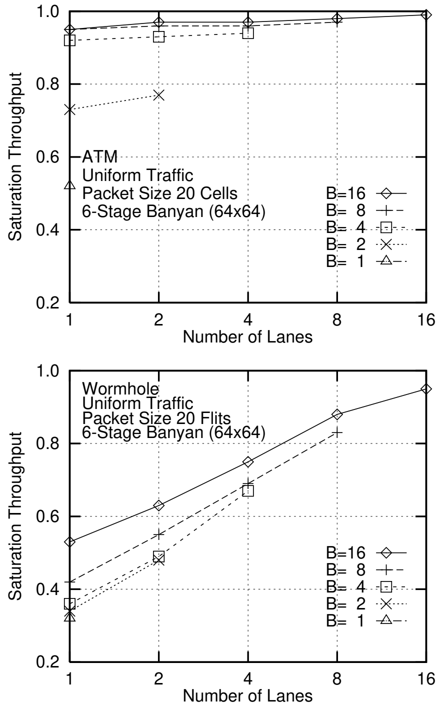

Figure 4: Saturation throughput versus lanes and buffer size: ATM (top), wormhole (bottom)

The measurements of figure 4 for wormhole routing and B=16 correspond to those in [Dally92] figure 14, n=6. For the comparison to be valid, our throughput values must be multiplied by 13/14 to account for our link capacity being 14/13 for wormhole (section 5); after this scaling, we observe the following. For single-lane wormhole, [Dally92] simulated input queueing –a blocked packet tail in a buffer also blocks the head of the next packet which is behind it in the same buffer– while we simulated shared buffering –no such blocking occurs; as a result, we obtained a (scaled) value of 0.49 versus a saturation throughput of only 0.34 in [Dally92]. For multi-lane wormhole, [Dally92] simulated input buffering (his figure 9(b)), again with head-of-line blocking inside each lane; our (scaled) saturation throughputs versus his values are:

$L = 2$: 0.58 versus 0.49 $L = 8$: 0.81 versus 0.78
$L = 4$: 0.70 versus 0.64 $L = 16$: 0.87 versus 0.85

This verifies the known property that shared buffering, which suffers no head-of-line blocking, performs better than input queueing/buffering, while, as we have shown in [KaVE95], their implementation cost is similar.

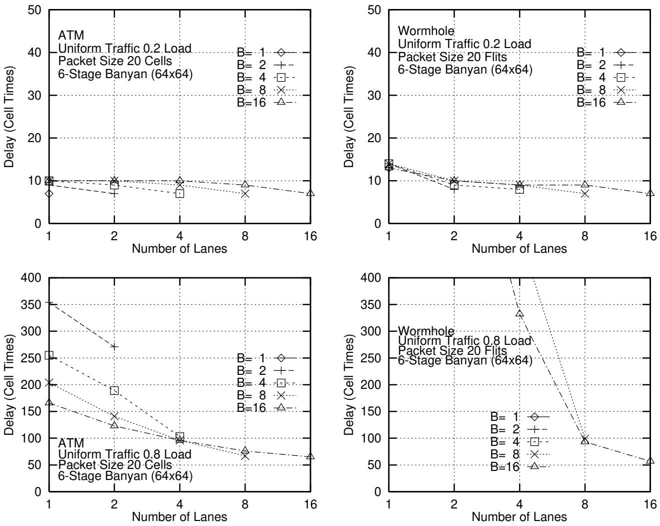

Figure 5: Delay versus lanes, for input load 0.2 (top) and 0.8 (bottom); ATM (left), wormhole (right)

Figure 5 plots delay for low input loads (0.2) and for high load (0.8). As seen, when the input load is much lower than the saturation throughput, the delay is quite low; delay grows very quickly when the load approaches the saturation point. Since ATM features higher saturation points than wormhole, ATM gives lower delays, in general.

6.2 Burst size dependence of performance

Burst size is an important parameter in network performance. Multiprocessor, multimedia, and other gigabit LAN sources often generate bursty traffic [Kung92]. In wormhole networks, long packets that span several switches in a network route block several other packets. To verify this hypothesis, we measured the saturation throughput of our wormhole network for packet sizes from a little shorter than one buffer size to a little longer than two buffer sizes. Figure 6 shows the result: as the packet size increases, performance drops more sharply when this size crosses boundaries of integer multiples of the per-switch, per-lane buffer size (8 flits).

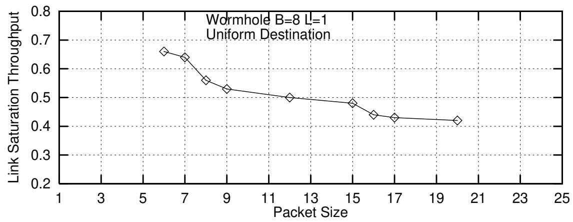

Figure 6: Packet size dependence; wormhole, B=8, L=1

Figure 7 plots saturation throughput for a few burst sizes in the range 0.5 to 8 Kilobytes (10, 20, 40, 160 cells or flits), when the buffer size is 8 cells or flits. In ATM, throughput stays high and is minimally affected, while in wormhole throughput decreases with increasing burst size, as predicted above. As in section 6.1, the number of lanes is not important for throughput in ATM, while it is in wormhole.

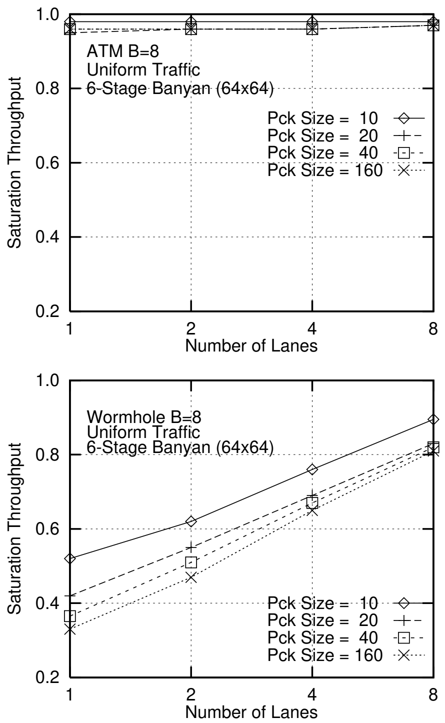

Figure 7: Saturation throughput versus lanes for various packet (burst) sizes: ATM (top), wormhole (bottom); B=8

Figure 8 plots delay for the same burst sizes, when the input load is 0.4. Contrary to the case of throughput, delay in ATM networks is sensitive to burst size. Furthermore, the number of lanes makes a difference now: for bursty traffic, multi-lane ATM provides significantly shorter delay than single-lane ATM (by a factor of three for 8 Kilobyte bursts in this case). When the number of lanes is large, delay is low for all burst sizes, in ATM. Wormhole exhibits longer delays; burst size and number of lanes affect delay dramatically, given that these factors affect considerably the saturation throughput, and delay is very sensitive to how close the network is to saturation.

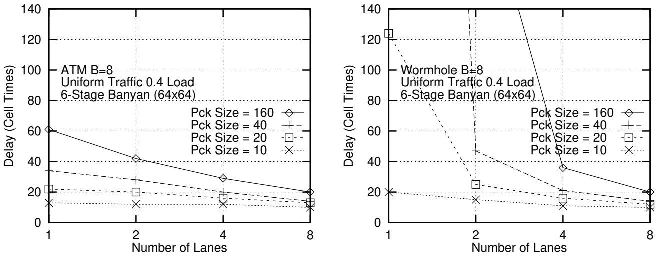

Figure 8: Delay for various packet (burst) sizes: ATM (left), wormhole (right); input load is 0.4; buffer size is 8

Another study that we conducted concerns *bimodal* traffic – a mixture of small and large packets, i.e. a network where short messages co-exist with large, bursty transfers. Under these circumstances, it is desirable that the short messages be minimally affected by the presence of the large bursts (more precisely, for a given total load, the ''short'' portion of the traffic should be minimally affected when the burstiness of the remaining traffic increases). In our experiments, half of the input load is due to packets of size 2 cells or flits, and the other half load is created by long packets; the size of the latter varies from 10 to 160 cells or flits (thus, short packets are 5 to 80 times more numerous than long packets). We measured delay under a (total) input load of 0.4 (0.2 short and 0.2 long packets), separately for the cells of short packets and for those of long packets. Buffer size per link is fixed at $B = 8$. Figure 9(top) plots the delays for the case of ATM. The delay of the cells in the long packets (right) behaves in an analogous way to the case where the entire traffic consists of long packets (figure 8 left). Here, the delays are shorter, because long packets constitute only half of the traffic, but the dependences and the shape of the curves remains the same. On the other hand, the delay of the cells in short packets (left) is *always short,* close to the value when all packets are short in figure 8, and this delay is affected very little by the length, i.e. the burstiness, of the other half of the traffic. This is precisely the desirable behavior for the network, as discussed above.

Figure 9(bottom) shows the corresponding plots for the case of wormhole routing. The two plots – short-packet cell delay (left) and long-packet cell delay (right) – resemble each other, and they are both similar to figure 8 (right) except that the values here are somehow lower than there, since the average packet length is lower here. As seen in figure 9 (left), the delay of the cells in the short messages is greatly affected by the length, i.e. the burstiness, of the other traffic in the network, which is an undesirable property of the wormhole protocol, contrary to the ATM one.

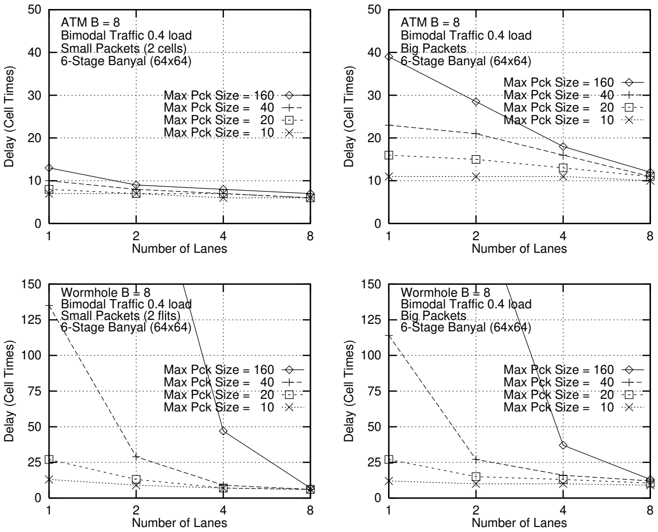

Figure 9: Delay of cells in short (left) and long (right) packets in ATM (top) and wormhole (bottom) when the size of the long packets varies; total input load is 0.4; B=8

6.3 Delay increase due to hot-spot traffic

Network performance under *non-uniform destination* traffic is very important in modern LANs. The traffic in gigabit LANs, especially when running multiprocessor and multimedia applications, is quite different from traditional network traffic that results from statistical multiplexing of many independent sources [Kung92]. Among others, in this new environment the number of sources or destinations may be small and sources may be dependent –for example, server computers or synchronization points in parallel applications easily create hot spots in the network.

In these experiments, we superimpose traffic going to one, two, or four particular hot-spot destinations over a background traffic of uniformly-destined 0.2 input load; the hot-spot traffic is equivalent to a load of 1.0 for each hot-spot (i.e., for this 64×64 network, the traffic load from source i to destination j is (0.2)/64 when j is not a hot-spot destination, and (1.0)/64 when j is a hot-spot). Thus, hot-spot traffic is saturated, while non-hot-spot traffic is relatively ''light''. When multiple hot-spots exist, we located them as ''far'' from each other as possible: $j = 0$ and $j = 32$

for two hot-spots, and 0, 16, 32, 48 for four hot-spots. Packet (burst) size is fixed at 20 cells or flits.

In an environment with hot-spots, the interesting measurement is the delay of network traffic directed towards non-hot-spot destinations. Traffic targeted to hot-spot destinations is bound to have high delay, due to output contention. In an ideal network, non-hot-spot traffic should remain unaffected by the possible existence of hot-spots (assuming that the two kinds of traffic are generated by independent processes, so that one can and should proceed when the other is blocked). Figure 10 shows the average non-hot-spot delay, as a function of the number of lanes, when zero, one, two, or four hot-spot destinations exist. Then, figure 11 further analyzes the non-hot-spot delay for the case of one hot-spot destination, in order to see how this delay varies for the different destinations, as a function of the *distance* from the hot spot.

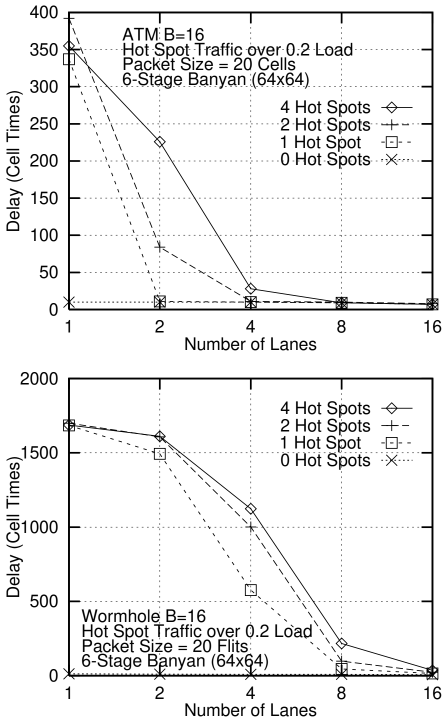

Figure 10: Delay of non-hot-spot traffic in the presence of 0, 1, 2, or 4 hot-spot destinations; ATM (top), wormhole (bottom); non-hot-spot load is 0.2, buffer space is 16

For the ATM protocol, figure 10 shows the following very clear behavior. The delay of non-hot-spot traffic is the same as in the no-hot-spot case (zero hot spots) when $L \geq 2$ in the one-hot-spot case, when $L \geq 4$ in the two-hot-spot case, and when $L \geq 8$ in the four-hot-spot case. In other words, ATM exhibits the *ideal* behavior when *the number of hot spots is less than the number of lanes.* This good behavior is a direct consequence of the way lanes are used in ATM (section 4): all cells going to one destination (e.g. hot-spot) are allowed to occupy at most one lane of buffer space, regardless of where they came from and which packet they belong to. Thus, as long as there are more lanes than hot spots, there will always remain at least one lane that is not blocked by hot-spot traffic; non-hot-spot cells can use that lane to overtake the congested traffic, suffering *no additional delay* due to this traffic.

Contrary to ATM, non-hot-spot traffic in wormhole networks suffers long delays even when the number of lanes is considerably higher than the number of hot spots. This is seen in figure 10 (notice that the vertical scale is 5 times larger than for ATM), and further in figure 11. It is due to the wormhole protocol allowing multiple packets with the same destination to occupy multiple lanes at the same time; thus, in our case, all lanes are easily blocked by (congested) hot-spot packets, dramatically delaying all other traffic as well.

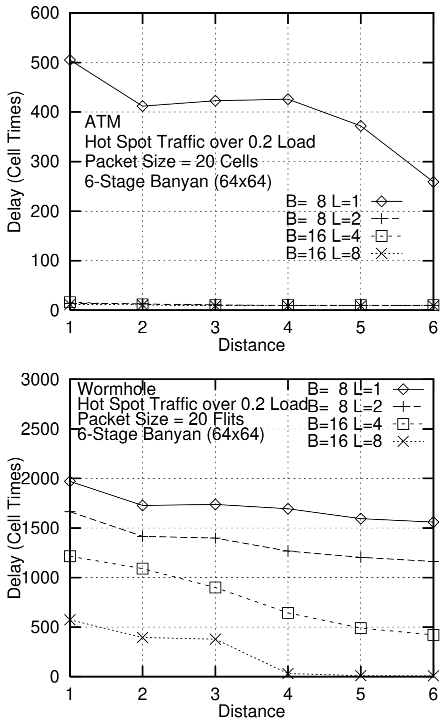

Figure 11: Delay versus distance from the (single) hot spot: ATM (top), wormhole (bottom)

It is quite interesting to understand the effect of hot-spot traffic on cells directed to *nearby* destinations. Intuitively, it is expected that destinations close to the hot-spot are affected more than destinations far from it. This is illustrated in Figure 11, plotting average cell delay to a destination D as a function of D's distance from the (single) hot spot H. Distance is defined as follows: given any network source S, consider the paths from S to D and from S to H;

distance is the number of network stages where these two paths are separate. As seen from the figure, for networks that have few lanes, where the presence of a hot spot significantly affects the remaining traffic, cell delay is indeed affected by the above distance, decreasing as the destination gets further away from the hot spot.

Figure 11 reveals one more phenomenon. In ATM networks with sufficient lanes for the given number of hot spots ($L \geq 2$), the non-hot-spot delay is uniformly low, *independent* of distance from the hot-spot. On the contrary, in wormhole networks, even when the lanes are many more than the hot spots (8 lanes and 1 hot spot), and thus the *average* non-hot-spot delay is low (45 cell times), the (few) destinations that are close to the hot spot suffer very high delay (400 to 600 cell times for the 7 destinations within a distance of 3 from the hot spot), while the rest of the destinations, which are far from the hot spot enjoy reasonable or low delays (32 cell times at distance 4, 11 at distance 5, and 9 cell times for the 32 destinations at distance 6). This demonstrates unfairness in the wormhole protocol, since the performance characteristics experienced by many destinations are affected by their location rather than by their traffic pattern.

Conclusions

ATM extended with credit-based flow control has notable similarities to wormhole routing, the popular multiprocessor interconnection network architecture. They both route packets after segmenting them into fixed-size units, called cells in ATM or flits in wormhole. Their implementation complexity does not differ much. And they both base flow control on credits and lanes, although they use lanes differently, which is the reason for their significant performance differences. Wormhole dedicates each lane to a single packet, until the end of that packet's transmission; if the packet gets blocked, so does the lane. Also, wormhole allows packets with the same destination to simultaneously occupy multiple lanes on the same link, thus depriving other traffic of the additional lanes.

We compared the performance of these two protocols, using simulation, with detailed and realistic models that are based on ATLAS I, a gigabit-link ATM switch chip that we are currently implementing. The results show that, for a same buffer size and number of lanes, ATM performs consistently better than wormhole –by a large margin in several cases. For ATM, saturation throughput approaches link capacity with buffer sizes as small as 4 or 8 cells per link (for 6-stage networks with short links), and stays high regardless of the number of lanes. On the other hand, multi-lane ATM switches provide much lower delay than single-lane ones when bursty or hot-spot traffic is present in the network. When the number of lanes is higher than the number of hot-spot destinations, non-hot-spot traffic remains virtually unaffected by the presence of hot spots. For wormhole routing, quite larger buffer sizes and many lanes are needed in order to achieve high saturation throughput. When saturation throughput is low, delay characteristics suffer as well, because any given operating point of the network is closer to saturation. Bursty and hot-spot traffic negatively affects wormhole performance to a much larger degree than it does for ATM.

In our view, ATM, which was originally conceived for wide area networking and was subsequently also used in the local area, is an appropriate technology for multiprocessor interconnection networks as well, especially in the form of networks of workstations, when extended with credit-based flow control.

Acknowledgements

This work was funded by the Commission of the European Union, through the "ASICCOM" project of the ACTS Programme. The ATLAS I chip is designed by Panagiota Vatsolaki, Chara Xanthaki, George Kalokerinos, George Kornaros, Dionisios Pnevmatikatos, and George Dimitriadis. Evangelos Markatos, Christoforos Kozyrakis, Vagelis Chalkiadakis, and many others have also helped in various ways. George Stamoulis, Costas Courcoubetis, Apostolos Traganitis, and the anonymous reviewers gave us useful comments on the paper. We thank them all a lot.

References

[ACPN95] T. Anderson, D. Culler, D. Patterson, and the NOW team: "A Case for NOW (Networks of Workstations)", *IEEE Micro Magazine,* vol. 15, no. 1, February 1995, pp. 54-64.

[Bork90] S. Borkar e.a.: "Supporting Systolic and Memory Communication in iWarp", *Proc. of the 17th Int. Symp. on Computer Architecture,* ACM SIGARCH vol. 18, no. 2, June 1990, pp. 70-81.

[CaVe96] J. Carbonaro, F. Verhoorn: "Cavallino: The Teraflops Router and NIC", *Proc. of the Hot Interconnects IV Symposium,* Stanford Univ., CA, USA, Aug. 1996, pp. 157-160.

[CSIM90] Mesquite Software, Inc.: "CSIM17 User's Guide" (see also: H. Schwetman: "CSIM User's Guide", Microelectronics and Computer Technology Corporation, March 1990).

[DaSe87] W. Dally, C. Seitz: "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks", *IEEE Trans. on Computers,* vol. C-36, no. 5, May 1987, pp. 547-553.

[Dally92] W. Dally: "Virtual-Channel Flow Control", *IEEE Tran. on Parallel and Distributed Systems,* vol. 3, no. 2, March 1992, pp. 194-205.

[Gall97] M. Galles: "Spider: A High-Speed Network Interconnect", *IEEE Micro,* vol. 17, no. 1, Jan./Feb. 1997, pp. 34-39.

[Inmo93] "IMS C104 Packet Routing Switch – Preliminary Datasheet", *INMOS Ltd,* Bristol, BS12 4SQ. UK, June 1993.

[KaVE95] M. Katevenis, P. Vatsolaki, A. Efthymiou: "Pipelined Memory Shared Buffer for VLSI Switches", *Proc. ACM SIGCOMM '95 Conference,* Cambridge, MA USA, August 1995, pp. 39-48. URL: ftp://ftp.ics.forth.gr/tech-reports/1995/ 1995.SIGCOMM95.PipeMemoryShBuf.ps.gz

[KaSS96] M. Katevenis, D. Serpanos, E. Spyridakis: "Credit-Flow-Controlled ATM versus Wormhole Routing", Technical Report FORTH-ICS/TR-171, *ICS, FORTH, Heraklio, Crete, Greece,* July 1996; URL: ftp://ftp.ics.forth.gr/tech-reports/1996/ 1996.TR171.ATM_vs_Wormhole.ps.gz

[KaSV96] M. Katevenis, D. Serpanos, P. Vatsolaki: "ATLAS I: A General-Purpose, Single-Chip ATM Switch with Credit-Based Flow Control", *Hot Interconnects IV Symp.,* Stanford Univ., CA, USA, Aug. 1996. URL: ftp://ftp.ics.forth.gr/tech-reports/1996/ 1996.HOTI.ATLAS_I_ATMswitchChip.ps.gz

[Korn97] G. Kornaros, C. Kozyrakis, P. Vatsolaki, M. Katevenis: "Pipelined Multi-Queue Management in a VLSI ATM Switch Chip with Credit-Based Flow Control", *Proc. 17th Conf. on Adv. Res. in VLSI (ARVLSI'97),* Univ. of Michigan at Ann Arbor, Sept. 1997, pp. 127-144; URL: ftp://ftp.ics.forth.gr/tech-reports/1997/ 1997.ARVLSI.Pipe_MultiQueue.ps.gz

[Kung92] H. T. Kung: "Gigabit Local Area Networks: A Systems Perspective", *IEEE Communications Magazine,* vol. 30, no. 4, April 1992, pp. 79-89.

[KuBC94] H.T. Kung, T. Blackwell, A. Chapman: "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing", *Proceedings of the ACM SIGCOMM '94 Conference,* London, UK, 31 Aug. - 2 Sep. 1994, ACM Comp. Comm. Review, vol. 24, no. 4, pp. 101-114.

[LeBo92] J. LeBoudec: "The Asynchronous Transfer Mode: A Tutorial", *Computer Networks and ISDN Systems,* May 1992.

[OzSV95] C. Ozveren, R. Simcoe, G. Varghese: "Reliable and Efficient Hop-by-Hop Flow Control", *IEEE Journal on Sel. Areas in Communications,* May 1995, pp. 642-650.

[QFC95] Quantum Flow Control Alliance: "Quantum Flow Control: A cell-relay protocol supporting an Available Bit Rate Service", version 2.0, July 1995; http://www.qfc.org

[TaFr88] Y. Tamir, G. Frazier: "High-Performance Multi-Queue Buffers for VLSI Communication Switches", *Proc. of the 15th Int. Symp. on Computer Architecture,* ACM SIGARCH vol. 16, no. 2, May 1988, pp. 343-354.

[Zerr96] B. Zerrouk, V. Reibaldi, F. Potter, A. Greiner, A. Derieux: "RCube: A Gigabit Serial Links Low Latency Adaptive Router", *Proc. Hot Interconnects IV Symp.,* Stanford Univ., CA, USA, Aug. 1996, pp. 13-17; http://humanus.ibp.fr