Buffer Requirements of Credit-Based Flow Control when a Minimum Draining Rate is Guaranteed

Manolis Katevenis[1]

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
E-mail: katevenis@ics.forth.gr Tel.: +30 (81) 391664 Fax: +30 (81) 391661

HPCS '97, Chalkidiki, Greece, June 1997
4th IEEE Workshop on Architecture & Implementation of High Performance Communication Subsystems
URL: file://ftp.ics.forth.gr/tech-reports/1997/1997.HPCS97.drain_cr_buf.ps.gz

ABSTRACT: Credit-based flow control (FC) offers significant advantages for data communication over networks such as ATM. One concern is the buffer memory requirement of credit FC, which is high when static, per-connection buffer allocation is used. The memory size can be reduced by dynamically sharing the buffer among the connections. We formulate and prove a new upper bound for the dynamically-shared buffer size, which applies when the downstream network guarantees that it will never throttle a connection down below a given minimum "draining" rate. This upper bound is inversely proportional to the minimum draining rate, and is independent of the number of connections. The minimum draining rate can be guaranteed by giving to the traffic arriving at a local area through long distance links priority over locally-generated traffic in the use of a given fraction of the outgoing network capacity. In practical situations, the new upper bound can be made small, thus providing a new, additional method to reduce the buffer memory cost to a very small percentage of the link cost. Unlike other such methods, no link underutilization may occur and there is no rate ramp-up delay.

KEYWORDS: *credit-based flow control, switch buffer space, hierarchical WAN, traffic class priority, minimum draining rate guarantee, ATM.*

1. Introduction

Flow control is a central issue in all packet switched networks. Credit-based flow control (credit FC) has important advantages for data communication [Kung92]. Under credit flow control, data are never dropped, so (expensive) communication resources are never wasted by transmitting information that will eventually be discarded. Credit FC allows stable network operation at full throughput utilization, so low-priority traffic can "fill in" the entire available capacity. Also, credit FC allows each transmission to start at full-speed; throttling will only occur when necessary and to the degree that is necessary. This paper addresses credit flow control in general, but with ATM in mind as the immediate application.

An important concern with credit-based flow control is its apparent need for large buffer memories. Because the required buffer size is proportional to the link propagation delay, the need for large memories is particularly pronounced in wide area networks. Actually, the concern is the *combined* growth of buffer memory in proportion to both the link length and the number of connections (or packet destinations) passing through the link. However, this is only true for the worst-case approach, when *static* buffer allocation is used. It is

[1] also with the University of Crete, Department of Computer Science, Heraklion, Crete Greece.

possible to use less than the worst-case buffer size, *dynamically* sharing this reduced buffer space among the connections; two such methods have been developed by [OzSV95] and [KuBC94]. As discussed in section 2, this reduction of buffer memory can lead to underutilization of the link, or some data loss, or transmission-rate ramp-up delays, or a combination of these, but the sharing of the buffer is controlled so as to make such negative effects infrequent or minor.

In this paper, we formulate and prove a new upper bound for the buffer space that is occupied by credit-flow-controlled traffic, when the network guarantees a given minimum draining rate for this buffer. This upper bound is proportional to the link propagation delay and inversely proportional to the minimum draining rate, but it is *independent* of the number of connections (or packet destinations) that pass through the link. In practical situations, the new upper bound can be made much smaller than the worst-case, static-allocation bound.

This new upper bound provides an additional method to reduce the buffer space requirements of credit-based flow control using dynamic buffer sharing. Unlike the methods of [OzSV95] and [KuBC94], no link underutilization may occur, no data will be lost, and there is no rate ramp-up delay. The assumption of minimum draining rate can be satisfied in a straightforward way by exploiting the hierarchical organization of networks: cells arriving at a local area through long distance links should have priority over locally-generated cells in the use of a given fraction of the outgoing network capacity. Also, end-user systems should either guarantee that they will always accept incoming traffic at a rate higher than a given minimum, or they should have enough private buffer memory to implement their own higher level of credit flow control.

The method introduced in this paper is *complementary* to the other existing credit flow control methods with dynamic buffer allocation. The other methods use less buffer space than the worst-case maximum, but do so in a way that will rarely degrade performance. The present method proves that, if appropriate restrictions are placed on the way the network operates, the worst-case maximum is much smaller than in the general case. The restrictions proposed in this paper are reasonable and implementable. In a real network, a mixture of methods is possible, with each approach being applied to those cases where it fits best.

The rest of this paper is organized as follows. Section 2 reviews credit-based flow control. Section 3 presents the motivation for the new scheme and intuitively explains the new upper bound. Section 4 describes how the conditions for the new upper bound can be satisfied by exploiting the hierarchical structure of networks and their priority scheme. Section 5 defines precisely the conditions, formulates, and proves the new upper bound for the buffer space. Finally, section 6 describes how end-user systems that provide their own buffer memory can implement their own level of credit-based flow control, if they are not willing to meet the requirements of section 4.

2. Review: Credit-Based Flow Control

This section reviews credit-based flow control. Credit flow control, also called back-pressure, or hop-by-hop window, or virtual channel (circuit) flow control, dates back to Tymnet [Rind77], GMD-NET [Gies78], and Transpac [SiDa79] in packet switched networks; see also the survey [GeKl80]. Under credit flow control, data is only transmitted to the downstream neighbor if the transmitter knows that buffer space is available at the receiver. This is implemented as illustrated in figure 1, using a count of available receiver buffer space. This "credit count" is decremented every time a unit of data (e.g. ATM cell) departs and is incremented when a credit token is received; credits are sent back when buffer space becomes available.

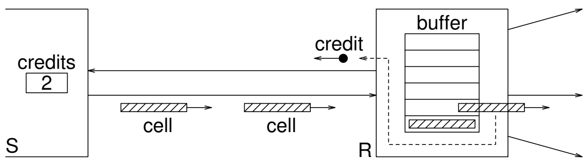

Figure 1: Credit-based flow control

For a given peak throughput, the necessary and sufficient buffer space at the receiver is equal to this *throughput* times the *round-trip time* (RTT) of data and credits. This buffer space, which is reminiscent of the braking or accelerating distance of a vehicle, is equal to the maximum number of cells and credits that can be in transit over the link and in the switch control logic. When *R*'s output suddenly stops, after full-speed operation, all these cells must be held in the given buffer space. Conversely, when

R's output suddenly starts, R must have that many cells available to sustain full speed operation until new cells have had the time to arrive from S.

In the simplistic *single-lane* credit FC scheme described above, the receiver buffer is indiscriminately shared by traffic going to all different destinations. When the flow of packets (cells) is not uniform, this indiscriminate sharing causes a behavior analogous to head-of-line blocking in input queueing [KaHM87]. Data going to congested paths or destinations move slower and thus accumulate in the buffers, hence blocking cells destined elsewhere from proceeding at full speed; the result is link underutilization and excess delays. An analogy in every-day life is single-lane streets (where the name came from): cars waiting to turn left block cars headed straight.

Multi-lane (per-VC) credit FC solves this performance problem by selectively allowing unblocked traffic to proceed, while preventing congested connections from occupying all available buffer space. The basic, worst-case approach is to dedicate separate buffer space –and hence to use a separate credit counter– for each connection (for each virtual circuit (VC) in ATM). The buffer space per connection must be equal to this connection's peak throughput times the round-trip-time (RTT). This *static* allocation and complete partitioning of the buffer space ensures complete independence of each connection from all others.

Actually, the number of partitions need not be as high as the number of connections –we only need as many partitions as the number of *flow groups.* A flow group, at each point in a network, is a set of connections that have a common destination and a common path to it, independent of their source. All members of a flow group can be flow-controlled together, because their data need never overtake each other –their common destination is assumed to be a single-threaded device, i.e. a device that does not process cells of one connection preferentially relative to those of another.

In a practical system, one can provide less buffer space than the above worst-case maximum; this reduced memory must be *dynamically shared,* using statistical multiplexing, among the flow groups. Figure 2 illustrates one simple such method, assuming the traffic consists of fixed-size cells (the unit of buffer space may be cells or bytes). The downstream switch has a buffer pool of size B; in order for this buffer not to overflow, the upstream switch maintains a credit count, *poolCr*. Separate credit counts, *fgCr*, are maintained for each flow group, initialized to a value b equal to the desired peak throughput of the flow group times RTT. A cell belonging to flow group i can depart if and only if $fgCr[i] > 0$ and $poolCr > 0$. When such a cell departs, $fgCr[i]$ and *poolCr* are decremented by 1, each. When that cell leaves the down-stream switch, a credit carrying its flow group ID, i, is sent up-stream. When credit i is received, $fgCr[i]$ and *poolCr* are incremented by 1, each.

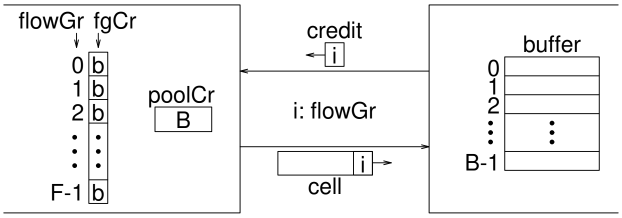

Figure 2: Simple dynamic sharing of buffer space B among F flow groups (L=B/b lanes)

Since at most b cells of a given flow group are allowed in the buffer at once, cells of at least $L = B/b$ different flow groups can always fit in the buffer at the same time; we call L the number of *lanes.* If more than L flow groups are active, though, it may happen that cells belonging to slow-moving (downstream congested) flow groups occupy a large portion or the buffer, thus leaving little space for the fast-moving flow groups; if the latter get less space than their throughput times RTT, the link will be underutilized. This dynamic buffer sharing method is simple enough to be implemented in hardware, at high speed; in fact, the *ATLAS I* single-chip ATM switch that we are currently developing [KaSV96] does so, for short-distance links ($b = 1$), at the rate of 50 Mega-cells per second. The above simple protocol is also reminiscent of the QFC protocol [QFC95]; *poolCr* corresponds to QFC's $Limit_{link}$, and $fgCr[i]$ corresponds to QFC's per-connection $Limit[i]$.

Better methods than the above simple dynamic sharing of the buffer have been developed, which are appropriate for long links and software (e.g. microcontroller) implementation. One of them, [OzSV95], reserves a certain minimum amount of buffer space, $b_{\min}$, for each flow group, and shares the remaining memory among all groups, still limiting each flow group to at most b cells; hysteresis is provided, which effectively reduces b when buffer space is scarce. As with the simple dynamic scheme of figure 2, underutilization of link throughput may still occur, but the hysteresis reduces its intensity or frequence, and at least no flow group will ever be completely starved. Another method, [KuBC94: section 6], allocates buffer space to each connection in proportion to its current use of throughput, plus an additional allotment to allow for the connection to ramp up its rate of transmission. Again, underutilization of the link (or some data loss) may occur sometimes, but this is presumed to be infrequent; rate ramp-up can only be exponential – not instantaneous – with the time-constant being proportional to the link round-trip time.

3. Motivation and Intuitive Derivation

This section provides an intuitive introduction to the proposed method for lowering the upper bound of buffer space for credit-based flow control. For motivation, we start with a discussion of cost.

3.1 *Memory Cost versus Transmission Cost*

Information storage (buffer memory) is much less expensive than information transmission [Kate87] [Fras95]: the cost of one lane of buffer memory −throughput times round-trip-time (RTT)− is much lower than the cost of the network link to which it corresponds. Both the link cost and the memory cost are proportional to the length of the link (because memory is proportional to RTT), so this length can be factored out. Both costs are also roughly proportional to throughput; as technology progresses, both link throughputs and memory chip capacities for a given cost grow. So, the relative cost of these two network components is about constant. For long-distance lines, Fraser [Fras95] estimates that the line cost is roughly *one third of a million times* higher than the cost of one lane of memory[2]! In other words, an improvement of the utilization of a WAN link by just 1 percent, or a corresponding reduction in the number of dropped cells, is worth as much as 3 thousand lanes of buffer space. Consequently, a credit flow control scheme that requires buffer size of a few hundred or thousand lanes has negligible or small relative cost. Only schemes that require tens or hundreds of thousands of lanes should be considered as expensive.

[2] a 45 Mbps long-distance line of a network operator costs about $45 per mile per month; its throughput-RTT product is 79 bytes/mile. At a memory cost of $1000 for 16 Megabytes of DRAM, one lane of buffer memory costs 0.5 cents per mile or 0.014 cents per mile per month (36-month write-down period). This one-lane memory cost is 330,000 times less than the line cost.

3.2 *What exactly is the Worst Case?*

Consider a credit-flow-controlled link of capacity C_{tot} and round-trip time T_{rt}. If a single flow group passes through the link, one lane of buffer space, $C_{tot}T_{rt}$, is enough; this allows the single flow group to reach a peak throughput equal to the link capacity C_{tot}. Now consider F flow groups in the link, with throughputs C_0, C_1,... C_{F-1} each. To sustain throughput C_i, flow group i needs a buffer portion of size C_iT_{rt} . Since the sum of all C_i's is C_{tot}, the sum of all these buffer portions is again $C_{tot}T_{rt}$ (a single lane). Why then is the required worst-case buffer space F lanes ($F \cdot C_{tot}T_{rt}$)? When and how does the worst case occur?

The worst case is when the following phenomenon is repeated for all flow groups, one after the other. A given flow group, α, is the only active one, and uses up all the available throughput, C_{tot}. Suddenly, flow group α is stopped *completely*, downstream from this buffer; consequently, data of size $C_{tot}T_{rt}$ that belong to α remain blocked in the buffer under consideration. Then, another flow group, β, becomes the only active one, using C_{tot}; when it gets stopped, data of size $C_{tot}T_{rt}$ belonging to β remain blocked in the buffer, and so on for all flow groups. In this way, $F-1$ flow groups may remain stopped, occupying $(F-1)C_{tot}T_{rt}$ buffer space, while the F'th flow group needs another $C_{tot}T_{rt}$ of buffer space in order to be able to use up all the available throughput (C_{tot}).

To say the same thing from another view point, the worst case is when $F-1$ of the F flow groups are completely stopped, downstream from this link, and yet they each want to reserve the right to suddenly re-start, at the peak rate C_{tot} , assuming all other flow groups will be stopped at that time. In order for this right to be guaranteed, and given that we do not know in advance which flow group will suddenly start or stop, the buffer under consideration must contain a "reserve" of $C_{tot}T_{rt}$ amount of traffic for *each* of the $F-1$ stopped flow groups.

The key requirement that leads to the above worst-case buffer size is that each and every flow group should have the right to (suddenly)[3] fluctuate its rate between two extreme values, zero and C_{tot}, whose ratio is infinite. If, instead, these two extremes have a bounded, "reasonable" ratio, then the worst-case buffer requirement is different, as discussed below.

3.3 *"Leaky" Buffer Portions*

In the above set-up of section 3.2, assume that no flow group is allowed to completely stop receiving traffic downstream from the link under consideration. To put it precisely, assume that any flow group is guaranteed to accept traffic at a rate of at least C_{min} if the link under consideration wishes to direct that much traffic to it. Consequently, if the buffer under consideration contains cells (traffic) belonging to flow group i, these cells will be departing from the buffer at a rate of at least C_{min} (provided the total output capacity of the buffer, for all flow groups, is not exceeded). Under these circumstances, the worst-case buffer occupancy of section 3.2 can never be reached if F is larger than C_{tot}/C_{min} .

Call B_i the amount of space in the output buffer of our link which is occupied by cells belonging to flow group i. If this i'th *buffer portion* is non-empty, then it "drains", or cells "leak" from it, at a minimum rate of C_{min} . What is the maximum number of non-empty portions in the buffer? If there are C_{tot}/C_{min} or more non-empty portions, given that they each drain at a rate of at least C_{min} , then the total buffer draining rate is at least C_{tot} . Given that the filling rate of the buffer is at most C_{tot} (the link capacity), it follows that buffer occupancy cannot grow any larger. Each of the non-empty buffer portions has a maximum size of one lane ($C_{tot}T_{rt}$), because it contains traffic of a single flow group, which is flow-controlled using one-lane's worth of credits (it never needs more). In conclusion, the upper bound for the buffer size under the new assumption is C_{tot}/C_{min} lanes[4], as opposed to the old value of F lanes. In a real network, F (the number of destinations or VCs) can be many thousands, while C_{tot}/C_{min} can be made a few tens or hundreds, as discussed in section 4.2. According to the discussion of cost above (section 3.1), the new worst-case buffer size has a low cost −which is not true for the old one.

The upper bound for the buffer occupancy that was intuitively derived above will be proven in a more general context in section 5. To explain this context, section 4 discusses a typical network organization where the minimum-draining-rate assumption can be met.

4. Hierarchy and Priorities in Networks

Before proving the new upper bound for the buffer occupancy, in section 5, we discuss here a typical network organization where it can be applied, so as to motivate the precise context and assumptions for the proof.

4.1 *Hierarchical Network Structure*

Like all large systems, wide area networks have a hierarchical structure. For example, the traditional telephony hierarchy −from which WANs evolve− consists of regional centers, sectional centers, primary centers, toll centers, and end offices. Figure 3 shows a simplified example of two levels of the hierarchy in a WAN. There are three "local" areas, X, Y, and Z, each with its network, which are interconnected via long distance communication lines.

[3] the "speed" of fluctuation is reminiscent of the allotment of buffer space for ramp-up purposes in the method of [KuBC94, section 6]. This further stresses the complementarity of the two schemes. In [KuBC94, section 6] only a single long-distance link is considered, while we allow multiple long-distance links to interact arbitrarily (and "suddenly") in the way in which they share the fraction of local throughput over which they have priority.

[4] this buffer size equals the lane size times the maximum ratio of throughput fluctuation that the downstream network is allowed to cause. This is again reminiscent of the buffer space in [KuBC94, section 6], which is the lane size times the throughput ramp-up ratio allowed per MTI (measurement time interval). However, in our case, the real fluctuation of a flow group's throughput can be arbitrarily large, since a flow group is free to use less throughput than C_{min} , as long as this is dictated by the upstream and not the downstream portion of the network; this arbitrarily low throughput can suddenly increase all the way to C_{tot} . Also, our buffer size calculation is for the worst case, while the similar formula in [KuBC94, section 6] is a steady-state calculation: during transient periods, credit reallocation is delayed when the excess cells are slow to drain from the buffer −see [KuBC94, section 6.1.5].

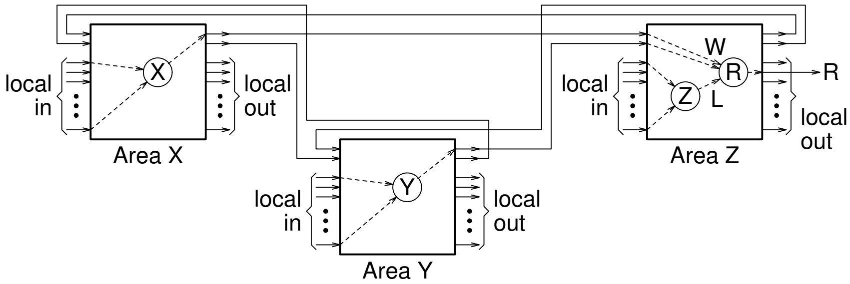

Figure 3: A WAN consisting of three interconnected LANs

Consider a typical end-user station, R, receiving traffic from various sources in the network. The circles in figure 3 represent, very abstractly, places where contention occurs among the cells that are destined to R. In this paper, we make no assumptions about how contention X or Y or Z is resolved, but we do make assumptions about contention R. The basic premise of our scheme is that by the time a unit of traffic (a cell) from areas X or Y has reached contention point R, this cell has traveled such a long distance that it is worth special status relative to cells originating from contention point Z. In other words, cells W, which have gone through a local-area selection process, have gained access to a long-distance link, and have traveled through that link, carry a considerable *investment* that the network has made in them, so they should be treated differently from cells L. Different treatment could mean higher priority, but that might lead to unfairness: long distance sources could monopolize R, driving local sources to starvation. Instead, we prefer something more flexible, as discussed below.

4.2 Priority over a Fraction of the Capacity

In the network example of section 4.1, call $C_{R,pk}$ the peak capacity of outgoing link R. Let $C_{R,min} = f \cdot C_{R,pk} > 0$ be a certain non-zero fraction f of that peak capacity of R. We postulate that *wide-area traffic* (marked W in figure 3) has priority over *local-area traffic* (marked L in fig. 3) over the fraction $C_{R,min}$ of R's capacity; also, R is not allowed to throttle its reception rate down below $C_{R,min}$. In other words, at contention point R, if traffic W asks for throughput not exceeding $C_{R,min}$, then it gets as much as it asks (and R is not allowed to refuse it e.g. by not issuing enough credits). If traffic W asks for throughput higher than $C_{R,min}$, then it gets at least $C_{R,min}$.

The fraction f is a network design parameter, and does not depend on the relative numbers of VCs in traffic classes W and L. Thus, the above priority scheme can introduce some unfairness. For example, if $f = 0.5$ and class W only contains one VC, while class L contains ten VCs, then the former VC will get 50% of R's capacity, while each of the latter will only get 5%. Yet, if f is relatively small, say 0.1, then it does not really matter if the VCs of class L lose this small fraction of traffic. Notice, however, that the receiving user station may want to throttle its total incoming traffic down to a value between $C_{R,pk}$ and $C_{R,min}$; then, traffic W will still have priority over capacity $C_{R,min}$, and, consequently, traffic L will be left with a smaller portion of the reduced influx of R.

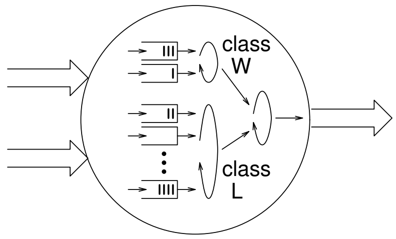

Figure 4: Example of implementing priority over a fraction of the capacity

There are various ways for a network implementation to satisfy the above priority requirement. Figure 4 illustrates one of them. Switches in a credit-flow-controlled network typically operate by visiting the VC queues in round-robin fashion [HaGa86] [Kate87]. If the simple two-level round-robin scheduling of figure 4 is used, then class W is guaranteed to get at least half of the outgoing capacity when it asks for it. Weighted round-robin [KaSC91] can also be used.

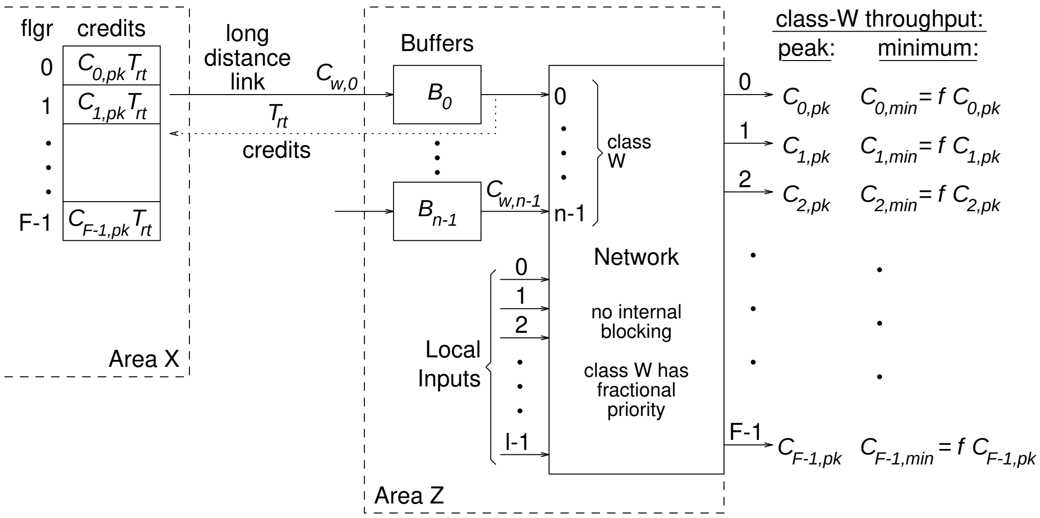

Figure 5: Network set-up for proving the new bound

A better implementation is to have a timer that counts ($1/C_{R,min}$) intervals. Every time a cell of class W is serviced, the timer is reset. Normally, the queues are serviced in (single-level) round-robin fashion (or weighted round-robin), independent of class. However, when the timer expires, a queue of class W is selected with top priority. This implementation also satisfies the requirement that class W will have at least capacity $C_{R,min}$ guaranteed for itself even when the receiving user station throttles down its overall reception rate (by not issuing enough credits). To satisfy the requirement, when the timer expires, if there is a non-empty queue of class W, a cell of that class is sent to R *even if* there is no credit for it, i.e. even if R risks to drop that cell.

5. A New Upper Bound for Buffer Occupancy

In this section, we formulate and prove the new worst-case bound for the buffer occupancy at the output of a long distance link in a hierarchical network like the one of section 4.1, under the priority assumption of section 4.2. Figure 5 shows the network used to derive our result.

There are n long-distance links entering into the local area Z under consideration. They each have a buffer, B_0,... B_{n-1}, at their outputs. Without loss of generality, we will compute the worst-case upper bound for the occupancy of buffer B_0. This buffer is fed by a long-distance link of capacity $C_{w,0}$ and round-trip time T_{rt}. The n buffers B_0 through B_{n-1} feed into the network of area Z, which is assumed to have no internal blocking, i.e. it can sustain any traffic pattern that does not violate the capacity limits of the input and output ports. If internal blocking exists, then the network has to be decomposed into stages, and the method of this section must be applied to these stages, in a hierarchical fashion.

The network has F outputs, numbered 0 through $F-1$. The network and its outputs guarantee the following property (*minimum draining rate guarantee*, section 4.2) concerning the traffic fed to the network from the long-distance buffers B_0 through B_{n-1} (called *class-W* traffic). Whenever there is class-W traffic, originating from whichever of B_0 through B_{n-1}, wishing to exit through output j of the network, then all of this traffic can enter the network and exit through output j if its total throughput does not exceed $C_{j,min}$, and at least $C_{j,min}$ of this traffic can do so when its total throughput exceeds $C_{j,min}$. For each of outputs 0 through $F-1$, there is a flow group of connections originating from area X. Flow group j, destined to output j, has enough credits ($C_{j,pk}T_{rt}$) at its departure from area X to sustain a peak throughput of $C_{j,pk}$ when no contention from other sources or back-pressure from destination j throttle it down to a lower value (destination j cannot throttle it down below $C_{j,min}$). The guaranteed lower threshold $C_{j,min}$ is a fixed fraction f of this peak throughput $C_{j,pk}$; f is fixed and the same for all outputs 0 through $F-1$. Under these circumstances, we will prove that:

$$B_0 \leq \frac{n_w}{f} \cdot C_{w,0} T_{rt} \quad (1)$$

where n_w is defined as:

$$n_w = \frac{C_{w,0} + C_{w,1} + \cdots + C_{w,n-1}}{C_{w,0}} \quad (2)$$

Here follows a list of the symbols used, with an explanation of what they stand for:

B_0: the occupancy of the buffer at the output of the long-distance link under consideration.

$C_{w,0}$: the capacity (peak total throughput) of the long-distance link under consideration.

T_{rt}: the round-trip time of traffic (cells) and credits, including credit processing time, on the long-distance link under consideration.

n_w: in the simple case where all incoming long-distance links, 0 through $n-1$, have the same capacity each, this is merely their number, n. In the general case, n_w is the peak total class-W throughput entering into the network of area Z, expressed as a multiple of the capacity $C_{w,0}$ of our long-distance link under consideration. To be precise, the capacities of the other long-distance links, $C_{w,1}$ through $C_{w,n-1}$, are measured at the output of the respective buffers, B_1 through B_{n-1}, while the capacity of our link 0 is measured on the link. The capacity at the output of a buffer should normally be slightly higher than the capacity of the corresponding incoming link, for the reason explained at the end of the proof. Equations (1) and (2), with the above interpretation of $C_{w,i}$, hold true when each buffer's output capacity is no less than the corresponding incoming link capacity.

f: the fraction of the output link capacities over which class-W traffic has priority (see discussion above).

j: an output of the area Z network, or the corresponding flow group from area X to that output; $0 \leq j \leq F-1$.

J: the set of all outputs j, or the set of their corresponding flow groups, for which some traffic (cells) exists in buffer B_0 at a given point in time.

$C_{j,pk}$: the peak throughput (capacity) of flow group j on the long-distance link under consideration (link number 0). The capacity of output j is $\geq C_{j,pk}$.

$C_{j,pk} T_{rt}$: the amount of credits that flow group j has on the long-distance link under consideration.

$C_{j,min}$: class-W traffic is always guaranteed to get throughput $C_{j,min}$ on output j whenever it asks for it; $C_{j,min} = fC_{j,pk}$.

In order to prove inequality (1), we will prove that if and when $B_0 \geq (n_w / f) \cdot C_{w,0} T_{rt}$, then the derivative of B_0 with respect to time is negative or zero at that moment, i.e. buffer B_0 is being drained faster than it is being filled, or at least its occupancy is not increasing. To simplify our calculations and the proof, we imagine the network traffic as a continuous fluid, and ignore the quantization of traffic and time into cells and time slots.

Imagine that, at some point in time, $B_0 \geq (n_w / f) \cdot C_{w,0} T_{rt}$. At that moment, consider the set J of all flow groups j for which some traffic (cells) exists in the buffer B_0. For each of these flow groups, the amount of traffic in the buffer B_0 cannot exceed the total amount of credits of that flow group on link 0, i.e. $C_{j,pk} T_{rt}$, so, by summing all these, we get

$$B_0 \leq \sum_{j \in J} C_{j,pk} T_{rt} \quad (3)$$

Combining this inequality with our assumption that $(n_w / f) \cdot C_{w,0} T_{rt} \leq B_0$, we get

$$(n_w / f) \cdot C_{w,0} T_{rt} \leq \sum_{j \in J} C_{j,pk} T_{rt} \quad (4)$$

Eliminating T_{rt} from both sides, expanding n_w from its definition (2), passing f to the right side and into the summation, and using $fC_{j,pk} = C_{j,min}$, we have

$$C_{w,0} + C_{w,1} + \cdots + C_{w,n-1} \leq f \sum_{j \in J} C_{j,pk} \quad => \quad (5)$$

$$C_{w,0} \leq \sum_{j \in J} C_{j,min} - (C_{w,1} + \cdots + C_{w,n-1}) \quad (6)$$

The sum $C_J = \sum_{j \in J} C_{j,min}$ is the total minimum guaranteed throughput for class-W traffic on all outputs in set J. Buffer B_0 contains traffic destined to *all* of the outputs in set J. If B_0 were the only source of class-W traffic, it could ask the network to forward to the outputs $j \in J$ any mix of traffic (cells) contained in B_0, and the network would be guaranteed to do so as long as B_0 asks for no more than $C_{j,min}$ to be forwarded to each $j \in J$. However, the other sources of class-W traffic, B_1 through B_{n-1}, have to be taken into consideration. In the worst case, all these other buffers are forwarding traffic (cells), at their peak output rates, $C_{w,1} \cdots C_{w,n-1}$, to the *same* outputs in set J where our buffer tries to forward its traffic, too. Also, in the worst case, the network may give priority to these other sources over our buffer B_0.

In such a worst case, a portion $(C_{w,1} + \cdots + C_{w,n-1})$ of the minimum guaranteed class-W throughput C_J of the outputs $j \in J$ is absorbed by the class-W traffic generated by buffers B_1 through B_{n-1}. Yet, there remains a portion of the minimum guaranteed throughput C_J that is not absorbed by the traffic generated by B_1 through B_{n-1}, and according to inequality (6) this portion is *no less than* $C_{w,0}$. Buffer B_0 is able to use up this remaining portion, because it contains cells destined to *all* of the outputs in set J, so it can choose any mix of them that is appropriate for using up the above remaining portion of output throughput (this choice is made automatically by the credit flow control mechanism of the area Z non-blocking network). Since buffer B_0 is able to empty itself at a rate no less than $\sum_{j \in J} C_{j,min} - (C_{w,1} + \cdots + C_{w,n-1})$, and since it is simultaneously being filled via its long-distance link at a rate not exceeding $C_{w,0}$, and since by inequality (6) the filling rate is no more than the emptying rate, it follows that the buffer's occupancy, B_0, cannot grow.

We have thus proven that whenever B_0 reaches or exceeds $(n_w/f) \cdot C_{w,0} T_{rt}$, this B_0 occupancy cannot grow any more. Since the buffer occupancy B_0 starts at zero when the network is empty, and since its derivative is non-positive for values of B_0 exceeding $(n_w/f) \cdot C_{w,0} T_{rt}$, it follows that this latter value is the upper bound of B_0, thus proving inequality (1).

For a practical system to operate stably, we must guarantee that the derivative will be negative when B_0 slightly exceeds the upper bound (1). This is guaranteed if the output capacity of B_0 is slightly higher than the capacity of its incoming link, $C_{w,0}$, as discussed above in the explanation of n_w.

Discussion

According to the new bound of inequality (1), the buffer memory required for each long-distance link in a wide area network is (n_w/f) lanes. In a usual hierarchical WAN, n_w is no more than a few tens. Also, the fraction f of output throughput over which class-W has priority can relatively easily be set in the range 0.5 to 0.1. The buffer requirements that result are on the order of a hundred lanes per long-distance link. According to section 3.1, this memory cost is negligible relative to the cost of the long-distance link (it is about 0.1 % of the link cost).

6. User-Level Buffering and Credit FC

According to our basic priority requirement, as formulated in section 4.2, the receiving user station R is not allowed to throttle down its reception rate of class-W traffic below $C_{R,min}$ (e.g. by not issuing enough credits). In some cases this may be acceptable, because $C_{R,min}$ may be a fairly low value, and the receiving device may be able to guarantee that it will always process arriving cells at least at that rate. In other cases, the receiving user station may prefer to provide its own buffer memory in order to implement pure credit-based flow control over the wide area at its own expense. This is an interesting concept, for the following reasons.

It is normally too expensive for the owner or the operator of a WAN to provide enough buffer memory for worst-case credit flow control in the sense of section 3.2, i.e. to provide $F \cdot C_{tot} T_{rt}$ buffer space. However, it may not be too expensive for the network *user* to provide his share of this buffer memory, and at least for *some users* this is certainly not expensive (e.g. supercomputers). Notice that each user station only needs to provide $C_{tot} T_{rt}$ buffer space for each of a relatively small number of "uncoordinated" long distance connections that it may wish to keep up at any given time. For example, a continent-wide connection with a propagation delay of 20 ms and a peak rate of 622 Mb/s requires 3 MBytes of buffer space. A high-end workstation with a 256 MB main memory, may easily wish to –*temporarily*– dedicate 5% of its memory to keep up 4 such active connections.

There is a second reason why user-level buffer and credit management can be desirable. In the above example of a continent-wide VC of 622 Mb/s peak rate, the throughput-RTT product is 3 MBytes. However, the user of the VC may know that the maximum possible burst size is, for example, 1 MByte (this may be the message size of the application). Then, the user can allocate just 1 MByte of buffer space, and issue the corresponding number of credits. Yet, for the network, this is a 622 Mb/s connection, and we want it to be that fast, in order to minimize the message transfer time.

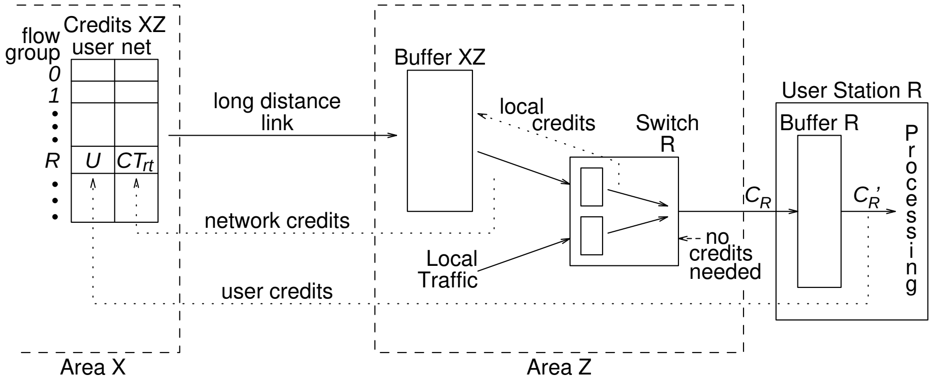

Figure 6: Hierarchical credit flow control: user-level buffering

If the network were to allocate worst-case buffer space for this connection, not knowing the user application that runs on it, this allocation would need to be 3 MBytes instead.

Figure 6 shows the *hierarchical credit* flow control protocol that must be used when the user station R wishes to provide its own memory, buffer R, in order for pure credit flow control to work at the WAN level with no restriction as to the minimum processing rate. At the sender end of the long distance link XZ, *two credit counters* are used, reflecting the two levels of flow control hierarchy that runs at that point. User credits refer to the space U in buffer R that has been reserved for the connection(s) from area X to R. Network credits refer to the network buffer XZ, and their purpose is to inform the XZ transmitter as to the progress that the cells of this flow group are making inside the area Z network. The number of user credits, U, depends on how much buffer space the user station wishes to allocate, and how large bursts it wishes to receive at no delay. The number of network credits is the peak rate of this flow group times the round-trip time T_{rt}. A cell of flow group R is allowed to depart when both credit values are positive; both counters are decremented upon the cell departure. When such a cell leaves buffer XZ, making progress inside the area Z network, a network credit for flow group R is sent back to the XZ transmitter, incrementing the corresponding counter. The user station R needs *not* send any "local" credits to switch R (at least for the flow group originating from X): we know that buffer R has room for every cell from X that reaches switch R, because that cell acquired a user credit at transmitter XZ. When station R processes a cell of this flow group and removes it from buffer R, it sends a user credit back to transmitter XZ.

The reception throughput C_R of station R as seen by the network stays always high, never dropping below $C_{R,min}$ as postulated in section 4.2. The cell processing throughput C_R' of station R may be arbitrary. In case C_R' drops to zero, buffer R will eventually fill up with cells, not being able to receive more. Still the requirement of section 4.2 is not violated, because buffer XZ and the area Z network will contain no cells destined to R (from area X). Such cells will have drained out of area Z into buffer R, or they are retained at transmitter XZ, where R's user credits are exhausted.

To summarize, the options for user station R are:

- to promise a minimum reception rate $C_{R,min}$ (plus whatever minimum we wish to set for receiving local traffic), and use any buffer size or no buffer at all. With this option, C_R must be larger than $C_{R,min}$; if C_R' falls below $C_{R,min}$ and inadequate buffering exists, then R may have to drop cells; or
- to make no promises on the processing rate C_R', but to provide its own buffer space and run a hierarchical credit flow control protocol as illustrated in figure 6. With this option, C_R' may be arbitrary; C_R will be set by the network; the amount of buffer provided by the user station determines the size of bursts that it may receive, and, together with the RTT of the connection, it determines the maximum average throughput of reception (bursts can be received at $C_{R,pk}$ instantaneous throughput).

In either case, the worst-case size of buffer XZ is as calculated in section 5, and independent of F.

Conclusions

We showed a new upper bound for the buffer requirements of long-distance links in wide area networks that use credit-based flow control. This new bound holds in hierarchical WANs when long-distance traffic is guaranteed at least a given share of throughput in the local area of the destination. By contrast to the traditional upper bound, which is proportional to the number of destinations (may be hundreds of thousands in real networks), the new bound is independent of that number, and is only proportional to the number of incoming long-distance links (usually no more than one hundred) and inversely proportional to the fraction of local throughput over which long-distance traffic has priority. Sections 4.1 and 4.2 showed the rationale and the realizability of our assumptions. Section 6 showed the relation of our assumptions to a concept that is interesting in its own right: hierarchical credit flow control and provision by the user of the buffer memory that is required for long-distance credit flow control.

The proposed new approach to credit-based flow control in wide area networks is complementary to previous methods of credit flow control in WANs with dynamic buffer sharing. Altogether, our conclusion is that credit flow control has a low relative cost in wide area networks, just like in local networks; given the important advantages that this flow control method offers, it should mature soon and be adopted. Our new bound on buffer space, the method by which it was derived, and the proposed network properties under which it holds, are steps in the above direction.

Acknowledgements

The author would like to thank Costas Courcoubetis, George Stamoulis, Apostolos Traganitis, and Vassilios Siris for useful discussions and comments on this paper, and the European Union ACTS (Advanced Communication Technologies and Services) Programme for the funding provided through the ASICCOM project.

References

[Fras95] A. Fraser: "Future Prospects for Wide Area Telecommunications", Keynote Speach at the *Hot Interconnects III Symposium,* Stanford Univ., CA, USA, Aug. 1995.

[GeKl80] M. Gerla, L. Kleinrock: "Flow Control: A Comparative Survey", *IEEE Trans. on Communications,* vol. 28, no. 4, April 1980, pp. 553-574.

[Gies78] A. Giessler, e.a.: "Free Buffer Allocation – An Investigation by Simulation", *Comput. Networks,* vol. 2, 1978, pp. 191-208.

[HaGa86] E. Hahne, R. Gallager: "Round Robin Scheduling for Fair Flow Control in Data Communication Networks", *Proc. IEEE Int. Conf. on Commun.,* June 1986, pp. 103-107.

[KaHM87] M. Karol, M. Hluchyj, S. Morgan: "Input versus Output Queueing on a Space-Division Packet Switch", *IEEE Trans. on Communications,* vol. COM-35, no. 12, December 1987, pp. 1347-1356.

[Kate87] M. Katevenis: "Fast Switching and Fair Control of Congested Flow in Broad-Band Networks", *IEEE Journal on Sel. Areas in Communications,* vol. 5, no. 8, October 1987, pp. 1315-1326.

[KaSC91] M. Katevenis, S. Sidiropoulos, C. Courcoubetis: "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip", *IEEE Journal on Selected Areas in Communications,* vol. 9, no. 8, October 1991, pp. 1265-1279.

[KaSV96] M. Katevenis, D. Serpanos, P. Vatsolaki: "ATLAS I: A General-Purpose, Single-Chip ATM Switch with Credit-Based Flow Control", *Hot Interconnects IV Symposium,* Stanford Univ., CA, USA, Aug. 1996. URL: file://ftp.ics.forth.gr/tech-reports/1996/1996.HOTI.ATLAS_I_ATMswitchChip.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", *Proc. of the ACM SIGCOMM '94 Conference,* London, UK, 31 Aug. - 2 Sep. 1994, ACM Comp. Comm. Review, vol. 24, no. 4, pp. 101-114.

[OzSV95] C. Ozveren, R. Simcoe, G. Varghese: "Reliable and Efficient Hop-by-Hop Flow Control", *IEEE Journal on Sel. Areas in Communications,* vol. 13, no. 4, 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. URL: http://www.qfc.org

[Rind77] J. Rinde: "Routing and Control in a Centrally Directed Network", *Proc. Nat. Comput. Conf.,* Dallas, Tx, USA, June 1977.

[SiDa79] J. Simon, A. Danet: "Controle des Ressources et Principes du Routage dans le reseau TRANSPAC", *Proc. Int. Symp. Comput. Networks,* France, Feb. 1979.