****

ATM Forum Document Number: ATM Forum/97-1085R1

****

Title: A switch algorithm for ABR multipoint-to-point connections

****

Abstract:
In a multipoint-to-point connection, multiple senders send data to the same destination. The multipoint-to-point connection can be implemented as a shared tree, where traffic from multiple branches is merged into the same stream after every merging point. If the same VPI/VCI values are used by all senders in the multipoint-to-point VC, it is impossible for the network to determine any sender-specific characteristics (such as the sender rate and whether it is bottlenecked at this link). Thus ABR multipoint traffic management becomes complex, since per-VC (or per-flow) accounting is no longer equivalent to per-source accounting. We design and simulate a switch scheme that achieves fairness among the senders in both point-to-point and multipoint-to-point connections.

****

Source:
Sonia Fahmy, Raj Jain, Rohit Goyal, and Bobby Vandalore
The Ohio State University
Department of Computer and Information Science

Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu http://www.cse.wustl.edu/~jain/

This work is sponsored in part by NASA Lewis Research Center.

****

Date: December 1997

****

Distribution: ATM Forum Technical Working Group Members (AF-TM)

****

****

1 Introduction

Multipoint-to-point connections require feedback to be returned to the appropriate sources at the appropriate times. The bandwidth requirements for a VC after a merging point is the sum of the bandwidths used by all senders whose traffic is merged (see figure 1). This is because the aggregate data rate after a merging point is the sum of all incoming data rates to the merging point [9].

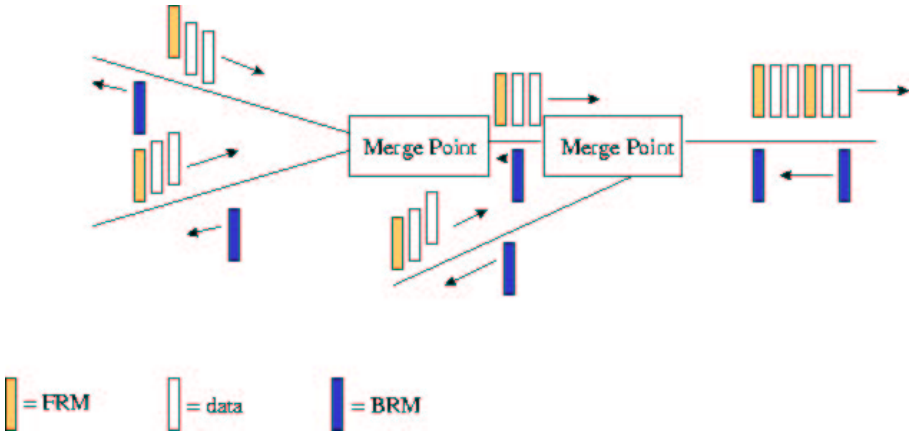


Figure 1: Multipoint-to-point connections

In [2], we defined different types of fairness for multipoint-to-point VCs implemented as shared trees. Among these, *source-based* fairness is the most preferred, since it is an extension of point-to-point fairness definitions [7]. To compute source-based fair allocations, a single N-to-one connection is treated as N one-to-one connections (VCs), regardless of which VC each source belongs to.

A source-based fairness algorithm must give the same allocation to all sources bottlenecked on the same link. Source-based fairness in VC merging implementations poses a number of problems, since sources in the same VC cannot be distinguished. The main considerations for switch algorithms in this case are to avoid any per-source accounting, and to avoid any attempt to estimate the effective number, or rates of active sources. However, this may increase the oscillations and transient response for many algorithms, since per-source accounting can improve switch algorithm performance.

In this contribution, we design a switch algorithm that gives source-based max-min fair allocations for point-to-point and multipoint VCs. The remainder of the contribution is organized as follows. First, we discuss the VC merging technique for avoiding cell interleaving in multipoint connections. Then, we summarize previous work on multipoint-to-point algorithms. In sections 4 and 5, we develop the rate allocation and merging point algorithms for multipoint connections. We analyze the performance of the algorithm by simulating it in a number of configurations in section 6.

2 VC Merging

The ATM adaptation layer (AAL) at each sender segments packets into ATM cells, marking the last cell of each packet. The AAL at the receiver uses the VPI/VCI fields and the end of packet marker to reassemble the data from the cells received. AAL5 does not introduce any multiplexing identifier

or sequence number in ATM cells. If cells from different senders are merged and interleaved on the links of a multipoint connection (implemented as a shared tree), the AAL5 at the receiver cannot reassemble the data. This is because all traffic within the group uses the same VPI/VCI. The identity of the sender is not indicated in each cell. Since the cells of different packets are interleaved, the packets get corrupted, as illustrated in figure 2.

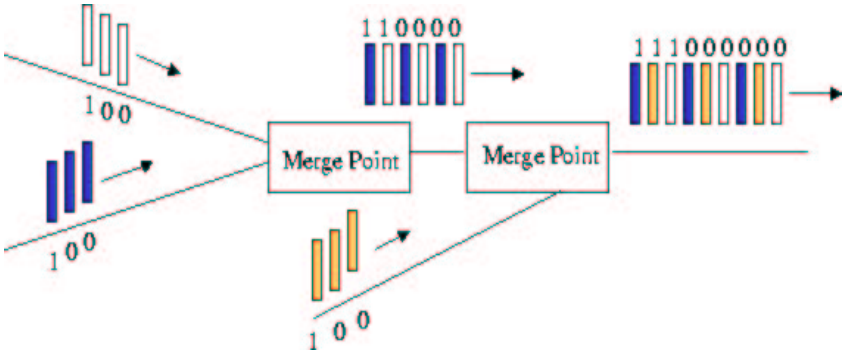


Figure 2: The cell interleaving problem

One of the solutions proposed to this problem is the VC merging approach. This approach buffers cells of other packets at the switch until all cells of the current packet go through (see SEAM [4] and ARIS [12]). A packet-based scheduling algorithm is implemented at the merging point, and separate queues are maintained for each flow. **A flow is defined as the cells of a VC coming on an input link.** The end-of-message bit signals to the switch that a packet from a different port can now be forwarded. See [13] for an analysis of VC merging. In this contribution, we focus on VC merging implementations, where cells from different senders are indistinguishable, since this is the most difficult case.

3 Previous Work

Traffic management rules for multipoint-to-point connections are still in their early phases of definition [5, 6]. Four different types of fairness have been defined for multipoint-to-point connections in [2]:

1. **Source-based fairness:** divides bandwidth fairly among active sources ignoring group memberships.
2. **VC/source-based fairness:** first gives fair bandwidth allocations at the VC level, and then fairly allocates the bandwidth for each VC among the active sources in the VC.
3. **Flow-based fairness:** gives fair allocations for each active flow, *where a flow constitutes the cells of a VC coming on an input link.*
4. **VC/flow-based fairness:** first divides the available bandwidth fairly among the active VCs and then divides the VC bandwidth fairly among the active flows in the VC.

Ren and Siu [10] have described an algorithm for source-based multipoint-to-point congestion control, which assumes that VC merging is employed. The algorithm operates as follows. When a

forward resource management (FRM) cell originating at a leaf is received at the merging point, it is forwarded to the root, and the merging point returns a backward resource management (BRM) cell to the source which had sent the FRM cell. The explicit rate in the BRM cell is set to the value of a register called MER (explicit rate), maintained at the merging point for each VC. The MER register is then reset to the peak cell rate. When a BRM cell is received at the merging point, the ER value in the BRM is used to set the MER register, and the BRM cell is discarded.

A better strategy is to maintain a bit at the merging point for each of the flows being merged [11]. The bit indicates that an FRM has been received from this flow after a BRM had been sent to it. Therefore, when an FRM is received at the merging point, it is forwarded to the root and the bit is set, but the RM cell is not turned around as in the previous algorithm. When a BRM is received at the merging point, it is duplicated and sent to the branches that have their bit set, and then the bits are reset. We implement this algorithm as explained in section 5, and show simulation results in section 6. In their papers [10] and [11], Ren and Siu only show simulation results for simple LAN configurations. We discuss more complex problems that arise in multipoint algorithms, and show simulation results for our proposed solutions.

4 Rate Allocation Algorithm

The rate allocation algorithm is employed at every switch to give the appropriate feedback to the senders [3]. The algorithm developed here is based upon the ERICA+ switch algorithm [8]. There are, however, a number of differences from ERICA+. We eliminate all the steps that required per-VC accounting. The reason for this is that we actually need *per-source* accounting. We avoid that for simplicity and scalability reasons, and for compatibility with VC merging switches. We first give the pseudocode of the algorithm, and then discuss some of the issues pertaining to rate allocation algorithms for multipoint connections.

4.1 Algorithm Pseudocode

The algorithm uses a measurement interval to measure the quantities required for computing the rate allocation. At the end of every interval, the algorithm averages and computes the quantities, which are used to give the appropriate feedback to the sources in the following interval. The algorithm measures the input rate and the available capacity, subtracting the capacity used by higher priority classes such as VBR. It also computes a function of the queueing delay and uses its value to scale the available capacity (in order to leave a percentage of the capacity for the queues to drain). See [8] for more details on the queue control concept and function. The ratio of the input rate to the target capacity is called the overload factor. The algorithm also uses the current cell rate of the sources, as indicated in the FRM cells. In addition, it keeps track of the maximum explicit rate given to sources during every interval.

In the pseudocode below, there are two options that are not necessary for the algorithm, but help reduce rate fluctuations in some cases (especially when the measurement interval value is very small). The first option (which we label option 1) does not use the most current current cell rate (CCR) value from FRM cells, but uses the maximum of the CCR values seen in FRMs in the current interval. This option is useful when there are multiple senders in the same VC, as explained in the

next subsection. The second option (option 2) uses exponential averaging for the maximum ER given in the previous interval.

The algorithm executes for each output port: when an FRM cell is received, when a BRM cell is received, and at the end of each measurement interval. Since the calculations of the input rate, target capacity and overload factor are the same as in the ERICA+ algorithm, we only briefly outline these here.

FRM cell is received for VC j:

(current cell rate)$_j$ = CCR field from the FRM cell

Or as an option (option 1: maximum CCR option):
IF (first FRM in interval)$_j$ = TRUE THEN
    (current cell rate)$_j$ = CCR field from the FRM cell
    (first FRM in interval)$_j$ = FALSE
ELSE
    (current cell rate)$_j$ = maximum (CCR field from the FRM cell, (current cell rate)$_j$)
END

BRM cell is to be sent out for VC j:

IF (overload factor $\geq 1+\delta$) THEN
    ER = (current cell rate)$_j$/overload factor
ELSE
    ER = maximum ((current cell rate)$_j$/overload factor, maximum ER in previous interval)
END

ER = minimum (target capacity, ER)

maximum ER in current interval = maximum (ER, maximum ER in current interval)

ER in BRM cell = minimum (ER, ER in BRM cell)

End of measurement interval:

target capacity = exponential average of target capacity across intervals (subtracting VBR capacity, and leaving capacity for queues to drain by using a queue control function as in [8])

input rate = exponential average of input rate across intervals

overload factor = input rate/target capacity

$\forall j$ (first FRM in interval)$_j$ = TRUE

maximum ER in previous interval = maximum ER in current interval

Or as an option (option 2: averaging the maximum ER in previous interval option):
maximum ER in previous interval = (1-α) $\times$ maximum ER in current interval + α $\times$ maximum ER in previous interval

maximum ER in current interval = 0

Notes:

1. The input rate, target capacity, overload factor, maximum ER in current interval and maximum ER in previous interval are computed and stored for each output port. The "first FRM in interval" and the "current cell rate" variables are stored for each VC for each output port.

2. In our simulations, the parameter δ is set to 0.1, and the parameter α is also set to 0.1. These are the recommended values in most cases.

3. The maximum ER previous averaging option (option 2) slightly reduces rate oscillations in some cases. It is not essential if its implementation complexity is high.

4. The maximum CCR option (option 1) also reduces rate oscillations, especially in cases of extremely small averaging interval values (< 200 μs for rates about 10 Mbps per source). It is not necessary in most cases. Exponentially averaging the maximum CCR values across intervals can further improve the performance. The next subsection discusses the usage of CCR in more detail.

4.2 Design Issues

Rate allocation algorithms for multipoint-to-point (or multipoint-to-multipoint) connections cannot distinguish cells from different sources in the same VC. Thus they cannot measure the rate of each source, or distinguish between overloading and underloading sources, or estimate the effective number of active sources. Such techniques are used in many point-to-point switch schemes, such as the MIT scheme, the UCSC scheme, and the ERICA scheme.

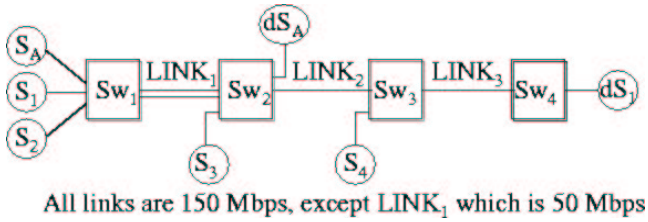


Figure 3: Example multipoint-to-point configuration with an upstream bottleneck

Furthermore, algorithms which use the CCR values noted from BRM cells will not work for multipoint-to-point connections. This is because it may be impossible to determine which source the RM cell belongs to. *The CCR value in the BRM cells at the merging point may not capture upstream bottleneck information for any of the flows whose traffic is being merged, since it may actually be the CCR of a downstream source whose bottleneck rate is high.*

Consider the following situation. Suppose a multipoint-to-point VC has two sources, one of which has a bottleneck rate of 58 Mbps, and the other has a bottleneck rate of 16 Mbps, and the two sources are being merged at a switch. Figure 3 shows an example where at $Switch_2$, S_1 (and S_2) of rate 16 Mbps and S_3 of rate 58 Mbps are being merged (we will simulate this case in sections 6.2.2 to 6.2.4). The source which is bottlenecked at 16 Mbps (say S_1) shares its bottleneck link with a point-to-point connection (S_A to dS_A). At the merging point, BRM cells of the higher rate source (the 58 Mbps source) are more frequently sent to *all* the sources in this VC being merged (assuming FRM cells were received from them) with an ER value that assumes the CCR is 58 Mbps.

This can result in overallocation to the lower rate source(s) being merged, and unfairness to the point-to-point connection.

Therefore, algorithms that use the CCR value for rate computation must use the value of the CCR indicated in FRM cells for computation when a BRM cell is received. This is the most up-to-date value of CCR anyway, since the CCR in the BRMs may be stale after traveling all the way to the destination and back. *The CCR value in the FRM cells at the merging point captures upstream bottleneck information for one of the flows whose traffic is being merged.* The FRM cells of the sources being merged, however, may still be indistinguishable at the merging point. In the remainder of this section, we argue that this does not significantly affect the convergence and steady state behavior of the algorithm.

Assume that there are two streams S_{low} and S_{high} being merged. Also assume that the CCRs of the two streams are 16 Mbps and 58 Mbps respectively. We will examine the situation when the forward CCR used to compute the ER for a stream is not the CCR corresponding to that stream.

When computing the ER for S_{low}, if the CCR of S_{high} (58 Mbps) is used, then the ER computed for S_{low} will be too high. But S_{low} is bottlenecked upstream of the merging point (otherwise its bottleneck rate will not be less than that for S_{high}, since S_{low} and S_{high} merge at the merging point and never split after that), so the ER given to S_{low} at the merging point will be overwritten by upstream switches.

For the case when the CCR of S_{low} is used to compute the ER for S_{high}, let us first consider the algorithm with the maximum CCR option. The only situation when the ER for S_{high} is calculated based upon the CCR for S_{low} is when only FRM cells of S_{low} have been seen since the beginning of the current interval (note that if no FRM cells have been seen at all, the CCR value used is the maximum seen in the previous interval, which will be the CCR of the higher rate source S_{high} unless S_{high} is sending at a very low rate, in which case the scheme should not allocate it high rates). Since S_{high} has a higher rate, it has a higher frequency of FRM cells, so it becomes highly improbable for this to hold. Assuming, however, that only S_{low} RM cells have been seen since the beginning of the current interval, the CCR divided by the overload factor will likely be smaller than the optimal rate for S_{high}. But this will eventually cause the overload factor to be less than $1 + \delta$, which means that the maximum ER allocated in the previous interval will be used in this case. The maximum ER in the previous interval captures the value for *all VCs* going to this output port, so fairness and high link utilization are ensured.

This argument can be extended for the algorithm without the maximum CCR option. In this case, instead of the smaller CCR being only used when no FRM cells from the higher rate source have been seen so far in this interval, it is the last FRM cell received that determines the CCR used. But, again, since the higher rate source has a higher FRM rate, it is statistically unlikely for the smaller CCR to be used. The maximum ER in the previous interval term ensures that if the small CCR is in fact used, the source is allocated at least as much as other VCs going to the same output port.

5 Merging Point Algorithm

This algorithm is based upon the multipoint-to-point algorithm developed by Ren and Siu in [11]. The algorithm is employed at every merging point where cells from different senders in the same multipoint-to-point VC are being merged and follow the same path to the destination. We first give the pseudocode for the algorithm, and then discuss some properties of the algorithm.

5.1 Algorithm Pseudocode

A flag (can be one bit) called Ready is maintained for each of the *flows* being merged. The flag indicates that an FRM cell has been received from this flow after a BRM cell had been sent to it.

Upon the receipt of an FRM cell from branch i:

1. Forward FRM cell to the outgoing link
2. Let Ready_i = TRUE

Upon the receipt of a BRM cell from the root:
FOR ALL upstream branches DO
    IF Ready_i = TRUE THEN
        Send a copy of the BRM to branch i
        Let Ready_i = FALSE
    END
END

When a BRM cell is about to be scheduled:
Perform the rate allocation algorithm as described in the previous section

5.2 Design Issues

There are other ways to implement multipoint-to-point ABR flow control algorithms. Each method offers a tradeoff in complexity, scalability, overhead, noise and response time.

In the above algorithm, a BRM cell is returned to a sender for every one or more FRM cells it sends. Thus the BRM to FRM cell ratio at the sender is less than or equal to one. In steady state, the ratio is likely to approach one, since the FRM rate and BRM rate will be similar. This is an important property of ABR flow control that should be maintained for multipoint-to-point connections. The BRM to FRM ratio *in the network* is also one in this case. This is contrary to the algorithm in [10] where FRM cells are turned around at merging points, and the same FRMs can be turned around at another merging point or the destination, creating many BRM cells that eventually get discarded in the network.

Also observe that in this scheme, since the merging point does not need to turn around every FRM cell, the overhead of the algorithm is significantly reduced. However, the scheme needs to duplicate BRM cells. With the new advances in multicast ATM switch architectures, this operation can be quite efficient.

The algorithm we use returns a BRM cell received from the root to the branches which have sent FRM cells to the merging point since the last BRM cell had been passed. This makes the scheme less sensitive to the number of levels of merging points, as compared to those schemes which turn around FRM cells (such as the scheme in [10]). This is because schemes turning around FRMs have to wait for an FRM to be received at every merging point, so their response time increases with the number of levels in the tree. In addition, the ER value returned by such schemes may be incorrect if no BRM cells have been received since the last one was sent, leading to rate oscillations and possibly large queue lengths.

6 Performance Analysis

This section provides a simulation analysis of the multipoint algorithm described in the previous two sections.

The results are presented in the form of four graphs for each configuration:

- (a) Graph of allowed cell rate (ACR) in Mbps versus time for each source
- (b) Graph of ABR queue lengths in cells versus time at the bottleneck port of each switch
- (c) Graph of link utilization versus time for each of the main links (those that connect two switches to each other)
- (d) Graph of number of cells received versus time for each destination

6.1 Parameter Settings

Throughout our experiments, the following parameter values are used:

1. Except where otherwise indicated (in sections 6.2.2 to 6.2.4), all links have a bandwidth of 155.52 Mbps (149.76 Mbps after SONET overhead is accounted for).
2. All multipoint-to-point traffic flows from the leaves to the root of the tree. No traffic flows from the root to the leaves, except for RM cells. Point-to-point connections are also unidirectional.
3. Except in section 6.2.3 where we experiment with the source parameter rate increase factor (RIF), we have set RIF to 1/32 in our simulations. We do not, however, expect the performance of the algorithm to be significantly influenced by the value of RIF, as seen in section 6.2.3.
4. The source parameter transient buffer exposure (TBE) is set to large values to prevent rate decreases due to the triggering of the source open-loop congestion control mechanism. This was done to isolate the rate reductions due to the switch congestion control scheme from the rate reductions due to TBE.
5. The queue control parameters are set as follows (see [8] for an explanation of these parameters). The two queue hyperbolic curve parameters a and b are set to 1.15 and 1 respectively.

The queue drain limit factor is set to 0.5 (which means that up to 50% of the link capacity can be used to drain queues), and the target queueing delay is set to 1.5 ms.

6. A fixed measurement interval is used to measure and average the input rate and available capacity, and to note the maximum allocation given (and possibly the maximum CCR value in FRM cells). The interval is set to 5 ms in all experiments except those in section 6.2.4.

7. Since we do not implement VC merging in our switches, we only use one cell long packets. Our next study will implement VC merging and examine its effect.

8. All sources are deterministic, i.e., their start/stop times and their transmission rates are known.

9. Simulation time is two seconds.

10. The simulations use both the maximum CCR option and exponentially averaging the maximum ER in the previous interval option as discussed in section 4. We have simulated all our configurations without using either option, and with each option separately, and the differences were insignificant. We do not show these results here for space considerations. In particular, the results when neither of the two options is enabled, and with extremely small measurement intervals (as with the simulations in section 6.2.4) showed that the algorithm still rapidly converges to the optimal allocations, and that the oscillations (though they do increase) were *not* significantly more than the results we show in section 6.2.4. From the discussion in section 4.2, however, we recommend the maximum CCR option for better performance.

6.2 Simulation Results

In this section, we discuss a sample of our simulation results. We mainly use two configurations, and experiment with different link lengths, initial cell rates of the sources, rate increase factor values, and lengths of the measurement interval.

6.2.1 Downstream Bottleneck Configuration

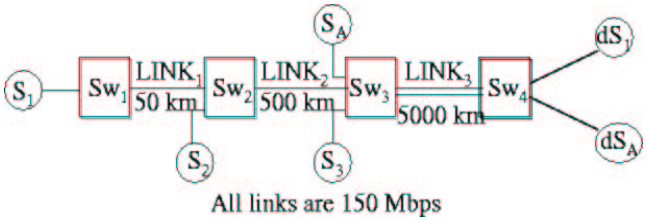


Figure 4: Example multipoint-to-point configuration with a downstream bottleneck

Figure 4 illustrates a configuration with two VCs: one of the VCs is a multipoint-to-point VC with three senders and one receiver, and the other is a point-to-point VC. Sources S_1, S_2, and S_3 are sending to destination dS_1, and source S_A is sending to destination dS_A. All links are approximately 150 Mbps (after SONET overhead is accounted for), and their lengths are as shown in the figure.

Clearly, all four sources are sharing a bottleneck link ($LINK_3$) between $Switch_3$ and $Switch_4$. *The aim of this example is to show the division of the 150 Mbps capacity of this bottleneck link among the sources.*

Applying the max-min fairness definition among sources, the optimal allocations should be:

$\{S_1, S_2, S_3, S_A\} \leftarrow \{37.5,\ 37.5,\ 37.5,\ 37.5\}$

Each of the four sources is allocated $\frac{1}{4} \times 150 = 37.5$.

Figure 6 illustrates the results of simulating the above configuration. The sources start with an ICR value of 25 Mbps, which is below their optimal allocation. Clearly, all sources rise to their optimal rates quickly (figure 6(a)), and the queues are small (figure 6(b)). The bottleneck link ($LINK_3$) is fully utilized (figure 6(c)). $LINK_2$ is 50% utilized (since only 2 of the 4 sources utilize it) and $LINK_1$ is only 25% utilized (1 out of 4 sources).

Observe that with source-based fairness, VCs that have a larger number of concurrently active senders get more bandwidth than VCs with less concurrent senders on the same link. This can be clearly seen from the slope of the cells received graph (figure 6(d)) for dS_1 and dS_A. Clearly dS_1 has a slope that is three times as large as that for dS_A. After 2 seconds, the ratio of cells received at dS_A to dS_1 is around 175000 to 520000, which is exactly 1 to 3. Thus the resource allocation is not max-min fair **among the VCs**. This means that the bandwidth allocated to a multipoint-to-point VC with N concurrent senders all bottlenecked on a certain link would be N times the bandwidth for a point-to-point VC bottlenecked on that same link, and N/K times that for a K-sender multipoint-to-point VC bottlenecked on the same link.

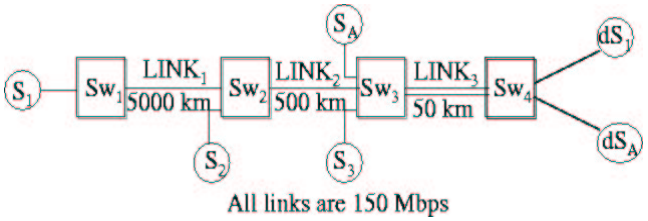


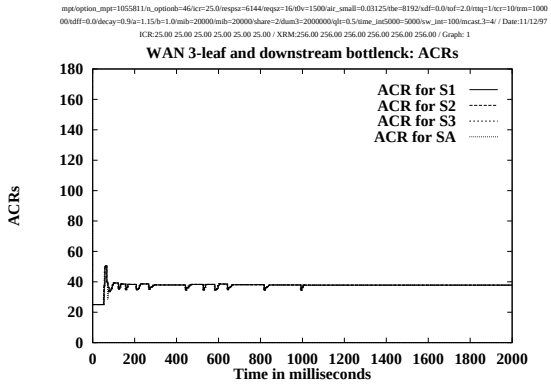
Figure 5: Example multipoint-to-point configuration with a downstream bottleneck

Figure 7 shows the situation with the same configuration but when $LINK_1$ is 5000 km and $LINK_3$ is 50 km, as shown in figure 5. There are slightly more rate fluctuations in this situation, due to the long feedback delay to the first source $S1$. The fluctuations are caused by variations in the value of the overload.

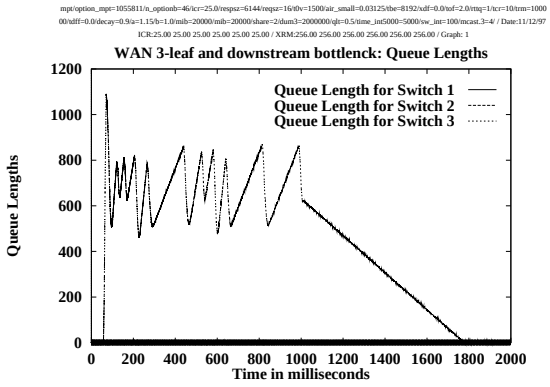
Figure 8 illustrates the results of simulating the first configuration (figure 4) when all sources start at a high ICR value. The ICR for all sources here is 100 Mbps. This creates an initial overload on $LINK_3$ of $\frac{400}{150} = 2\frac{2}{3}$. The algorithm recovers from this situation and all sources converge to the correct value of approximately 37.5 Mbps (figure 8(a)). The queues at $Switch_3$ start dropping after approximately one round trip (figure 8(b)).

Figure 9 shows the situation when $LINK_1$ is the longest link (figure 5) with a high ICR value (100 Mbps). The rates converge to about 37.5 Mbps (figure 9(a)), the queues are drained quickly (figure 9(b)), and the bottleneck link is fully utilized (figure 9(c)).

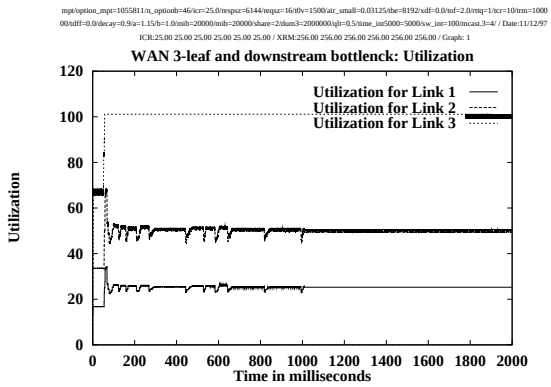
Figure 10 shows the results for the configuration of figure 4 (long $LINK_3$), but different sources



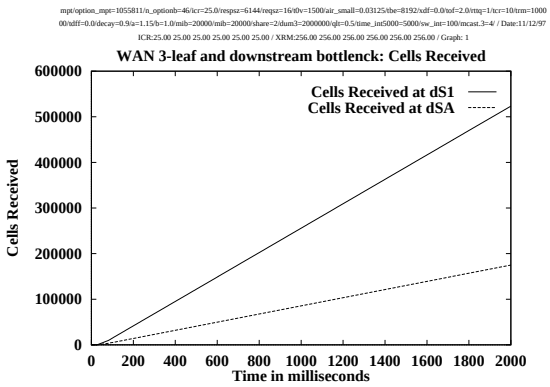
(a) Transmitted Cell Rate



(b) Queue Length

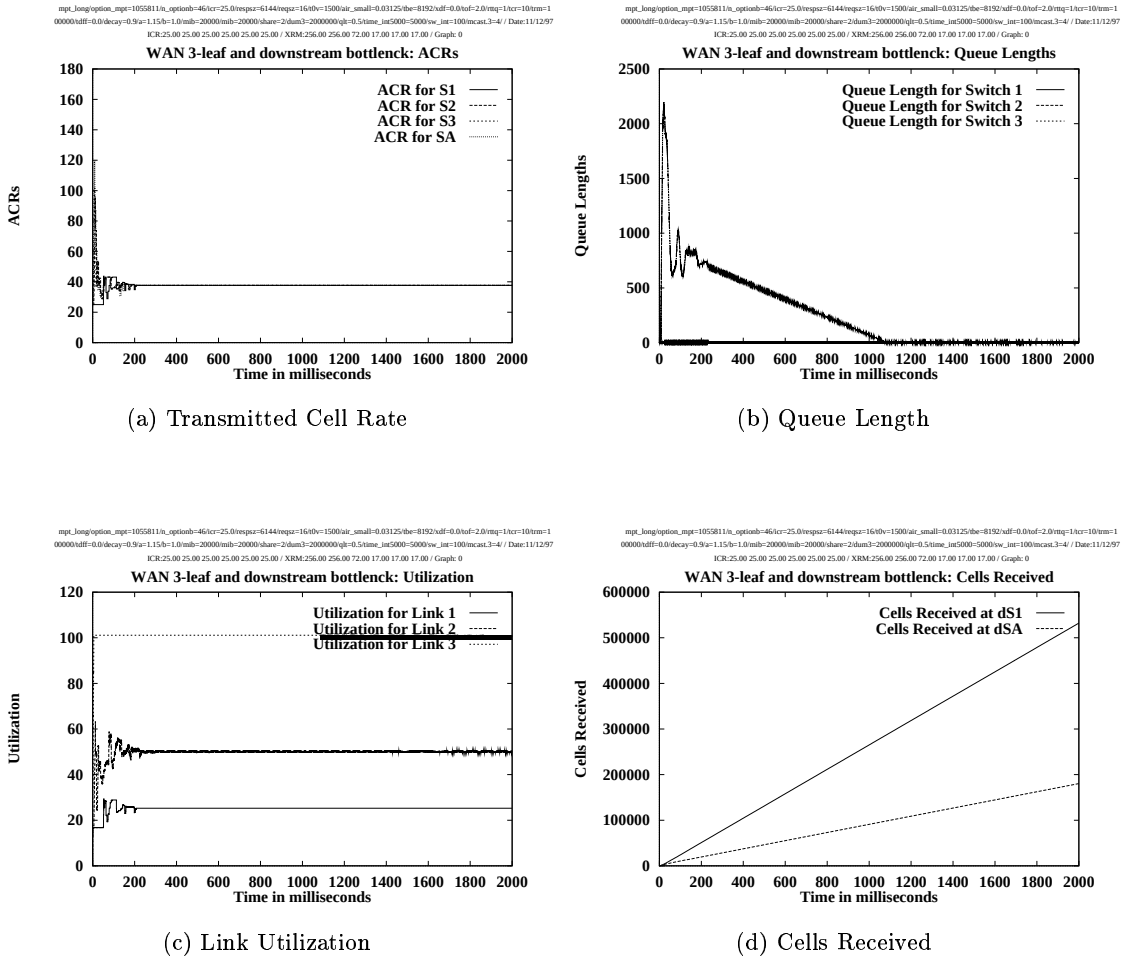


(c) Link Utilization



(d) Cells Received

Figure 6: Results for a WAN multipoint-to-point configuration with a downstream bottleneck (long LINK3, low ICR)



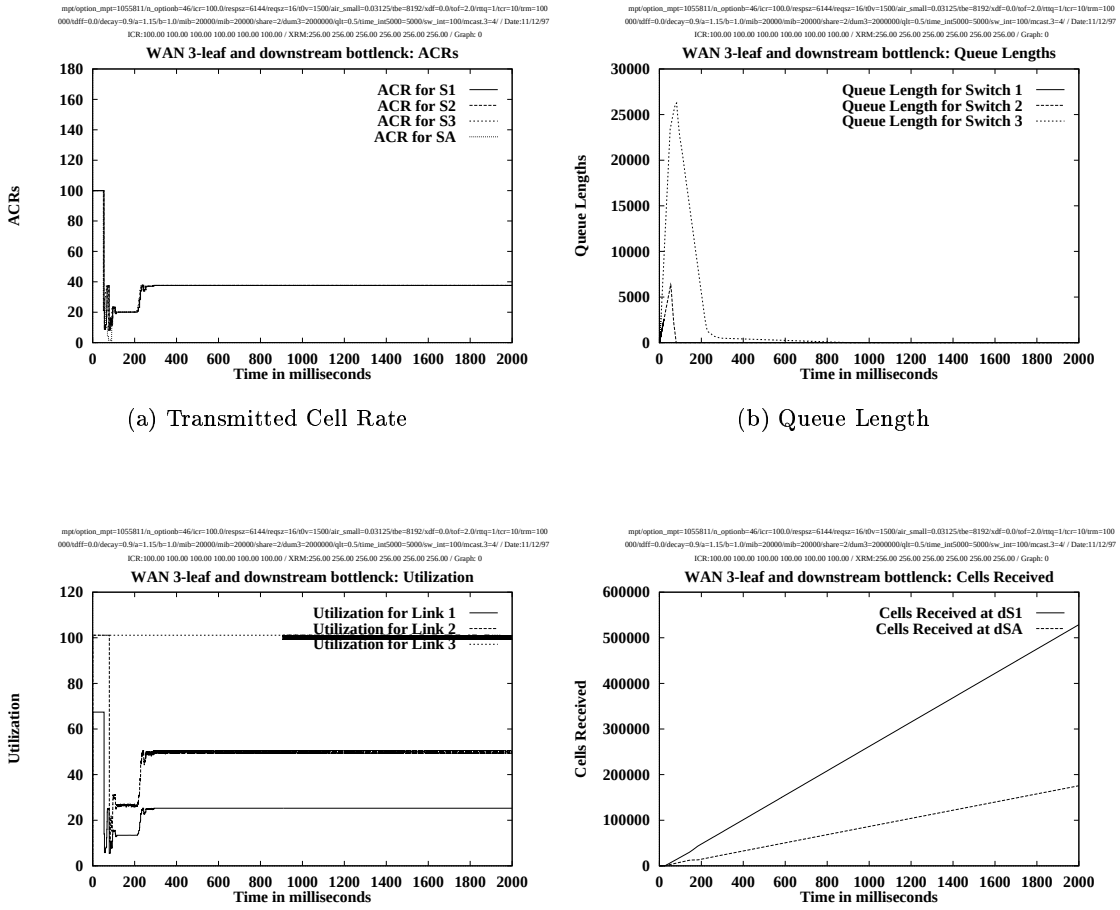
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 7: Results for a WAN multipoint-to-point configuration with a downstream bottleneck (long LINK1, low ICR)



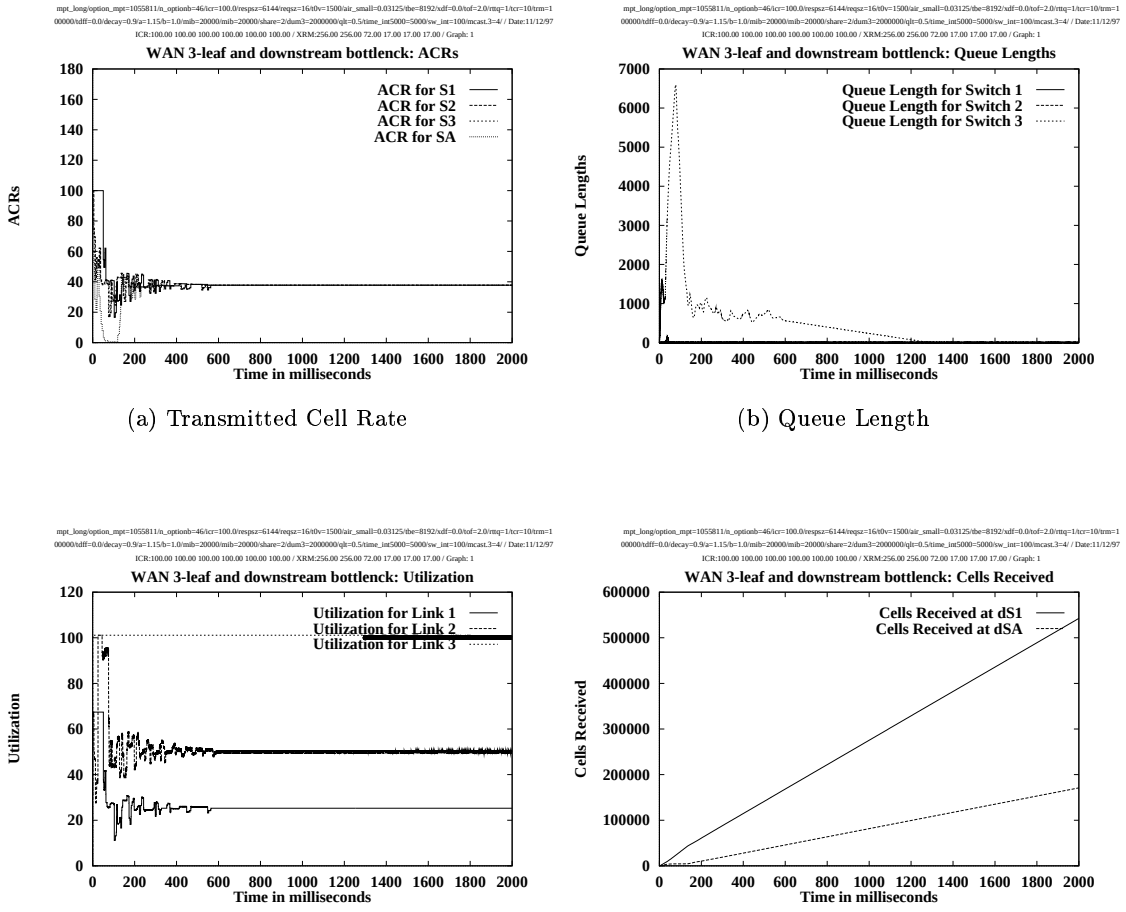
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 8: Results for a WAN multipoint-to-point configuration with a downstream bottleneck (long LINK3, high ICR)



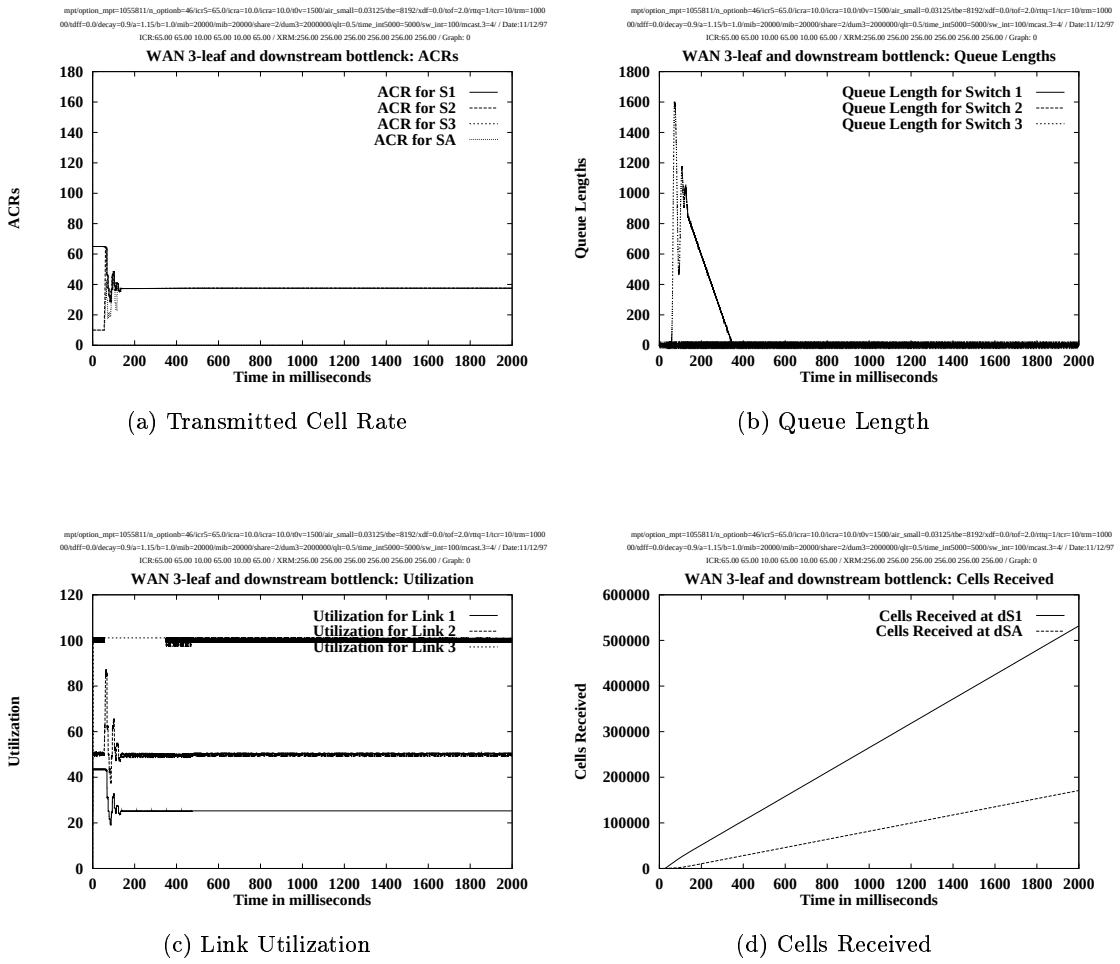
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 9: Results for a WAN multipoint-to-point configuration with a downstream bottleneck (long LINK1, high ICR)



(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 10: Results for a WAN multipoint-to-point configuration with a downstream bottleneck (long LINK3, different ICRs)

start at different ICR values. Sources S_1 and S_3 start at an ICR of 65 Mbps, while sources S_2 and S_A start at 10 Mbps. Notice that the sum of the source rates for all sources is 150 Mbps, so the initial load value is close to 1. The rates for sources S_1 and S_3 are quickly reduced, while those of sources S_2 and S_A quickly rise, as seen in figure 10(a). The queues are also quite small (figure 10(b)).

6.2.2 Upstream Bottleneck with Heterogenous Links Configuration

Figure 11 illustrates a configuration with two VCs: one of the VCs is a multipoint-to-point VC with four senders and one receiver, and the other is a point-to-point VC. Sources S_1, S_2, S_3 and S_4 are sending to destination dS_1, and source S_A is sending to destination dS_A. All links are approximately 150 Mbps (after SONET overhead is accounted for), except for the link between $Switch_1$ and $Switch_2$ ($LINK_1$) which is **only 50 Mbps**. The link lengths are as shown in the figure. Clearly, sources S_1, S_2 and S_A are bottlenecked at $LINK_1$, while sources S_3 and S_4 are

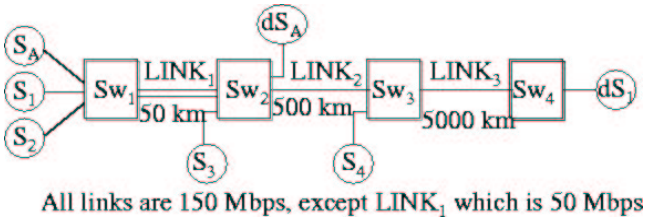


Figure 11: Example multipoint-to-point configuration with an upstream bottleneck

bottlenecked at $LINK_3$. *The aim of this example is to illustrate the allocation of the capacity left over by sources bottlenecked on* $LINK_1$ *to the sources bottlenecked on* $LINK_3$*, and the fairness of the allocation among the point-to-point and multipoint sources.*

The allocation vector according to the source based fairness definition is:

$\{S_1, S_2, S_3, S_4, S_A\}$ $\leftarrow$$\{16.67,\ 16.67,\ 58.33,\ 58.33,\ 16.67\}$

This is because each of sources S_1, S_2 and S_A is allocated one third of the bandwidth of $LINK_1$. At $LINK_3$, the $50 \times \frac{2}{3} = 33.33$ Mbps used by sources S_1 and S_2 is subtracted from the available bandwidth, and the remaining capacity (116.67 Mbps) is equally divided upon sources S_3 and S_4.

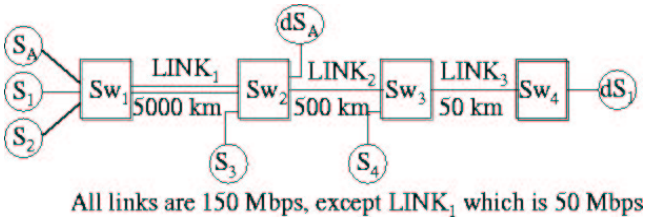


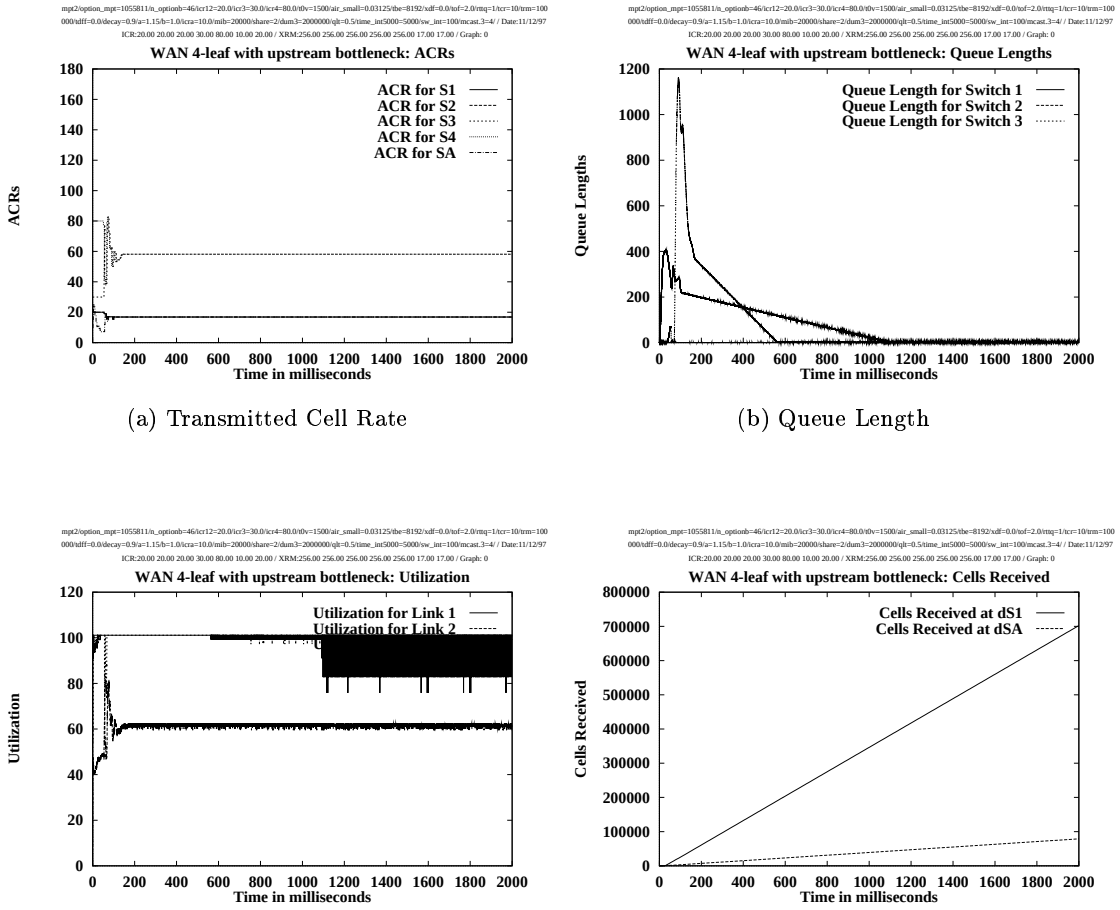
Figure 12: Example multipoint-to-point configuration with an upstream bottleneck

Figures 13 and 14 illustrate the results for the configuration when $LINK_1$ is of length 50 km and $LINK_3$ is of length 5000 km (figure 11), and when $LINK_1$ is of length 5000 km and $LINK_3$ of length 50 km (figure 12), respectively. Sources S_1 and S_2 start at an ICR of 20 Mbps. Source S_3 starts at 30 Mbps and source S_4 starts at 80 Mbps. Source S_A starts at 10 Mbps.

As seen in figures 13(a) and 14(a), sources S_1, S_2 and S_A converge to about 16.67 Mbps, while sources S_3 and S_4 converge to about 58.33 Mbps. The queues are bounded to reasonable values (figures 13(b) and 14(b)) and utilization of the bottleneck links ($LINK_1$ and $LINK_3$) are close to 100% (figures 13(c) and 14(c)). Destination dS_A gets much less throughput than dS_1 (figures 13(d) and 14(d)), since source S_A is bottlenecked on a 50 Mbps link with 2 other sources. After 2 seconds, the ratio of the throughputs for destinations dS_A to dS_1 is approximately 80000 to 700000 which is 0.11. The slopes of the two lines also have the same ratio. This is close to the optimal value since $16.67/133.09 = 0.13$.

6.2.3 Effect of Large Rate Increase Factor Values

The rate increase factor determines the maximum increase when a BRM cell indicating underload is received. If the RIF is set to a fraction less than one, the maximum increase at each step is limited to RIF $\times$ the peak cell rate for the VC. Setting RIF to small values is a more conservative



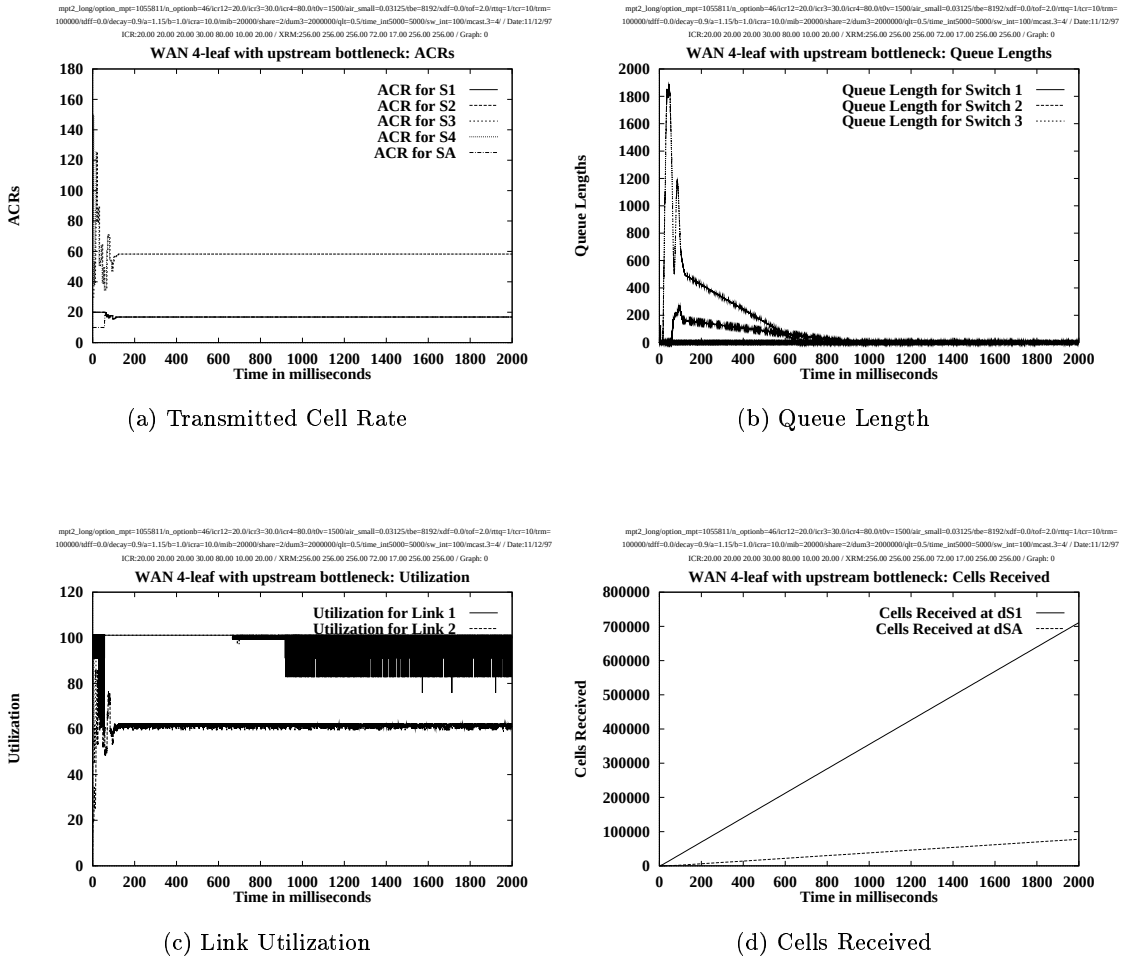
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 13: Results for a WAN multipoint-to-point configuration with an upstream bottleneck (long LINK3)



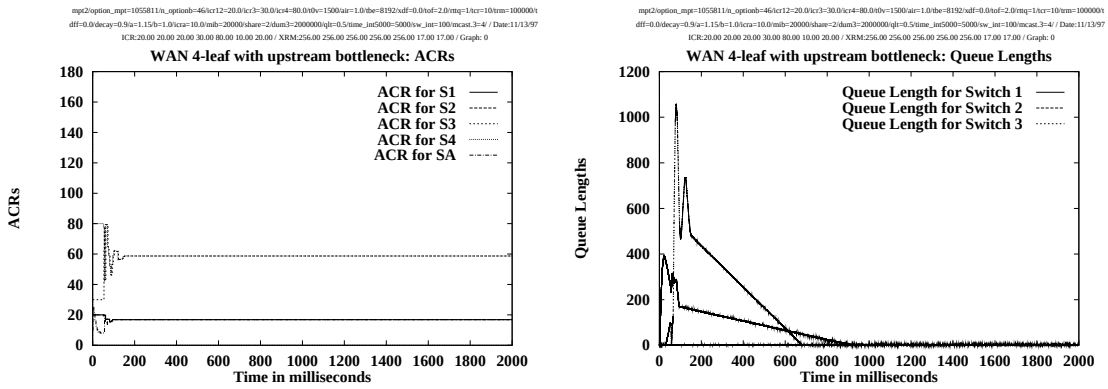
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

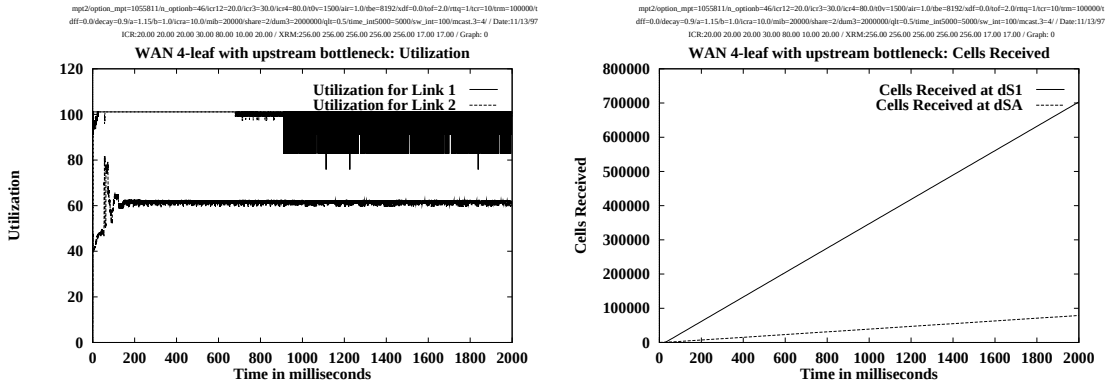
(d) Cells Received

Figure 14: Results for a WAN multipoint-to-point configuration with an upstream bottleneck (long LINK1)



(a) Transmitted Cell Rate

(b) Queue Length



(c) Link Utilization

(d) Cells Received

Figure 15: Results for a WAN multipoint-to-point configuration with an upstream bottleneck (long LINK3, large RIF)

strategy that controls queue growth and oscillations, especially during transient periods. It, however, may slow down the response of the system when capacity suddenly becomes available leading to underutilization.

Figure 15 illustrates the results for the configuration of figure 11 when the rate increase factor (RIF) is set to its maximum possible value, which is 1. Part (a) of the figure shows that the rates do not oscillate more than the corresponding figure with a small RIF value (figure 13(a)). The queues in figure 15(b) are also similar to those in figure 13(b).

6.2.4 Effect of Extremely Short Measurement Intervals

As discussed in section 4, extremely short measurement intervals can cause the algorithm to suffer from oscillations. To examine this effect, we have simulated the algorithm with a measurement interval of 200 μs. Recall that in the upstream bottleneck configuration (shown in figure 11), the

optimal rates for sources S_1, S_2 and S_A are 16.67 Mbps, and those for sources S_3 and S_4 are 58.33 Mbps. This implies that, in steady state, RM cells for sources S_1, S_2 and S_A arrive every:

$$\frac{Nrm \times bits/cell}{ACR} = \frac{32 \times 53 \times 8}{16.67\ M} = 813.92\ \mu s$$

For sources S_3 and S_4, RM cells arrive every:

$$\frac{Nrm \times bits/cell}{ACR} = \frac{32 \times 53 \times 8}{58.33\ M} = 232.61\ \mu s$$

Setting the measurement interval to 200 μs means that RM cells for S_3 and S_4 might not be received every measurement interval, and that RM cells for S_1, S_2 and S_A might not be received for 4 consecutive measurement intervals.

In order to receive at least one FRM cell from the highest rate source in a certain interval, the interval length should be $> \frac{Nrm}{ACR\ maximum}$. This condition is likely to hold for reasonably long intervals, unless *all* sources are sending at very low rates, in which case the overload factor will be low and their rates will increase if they have data to send.

Figure 16 illustrates the results for the configuration of figure 11. Clearly, the short averaging interval causes more oscillations, but the rates of the sources still converge to their max-min fair rates. Also observe that the number of cells received for both connections is the same as in figure 13(d). Increasing the value of the parameter α (in section 4) can reduce the oscillations.

7 Summary

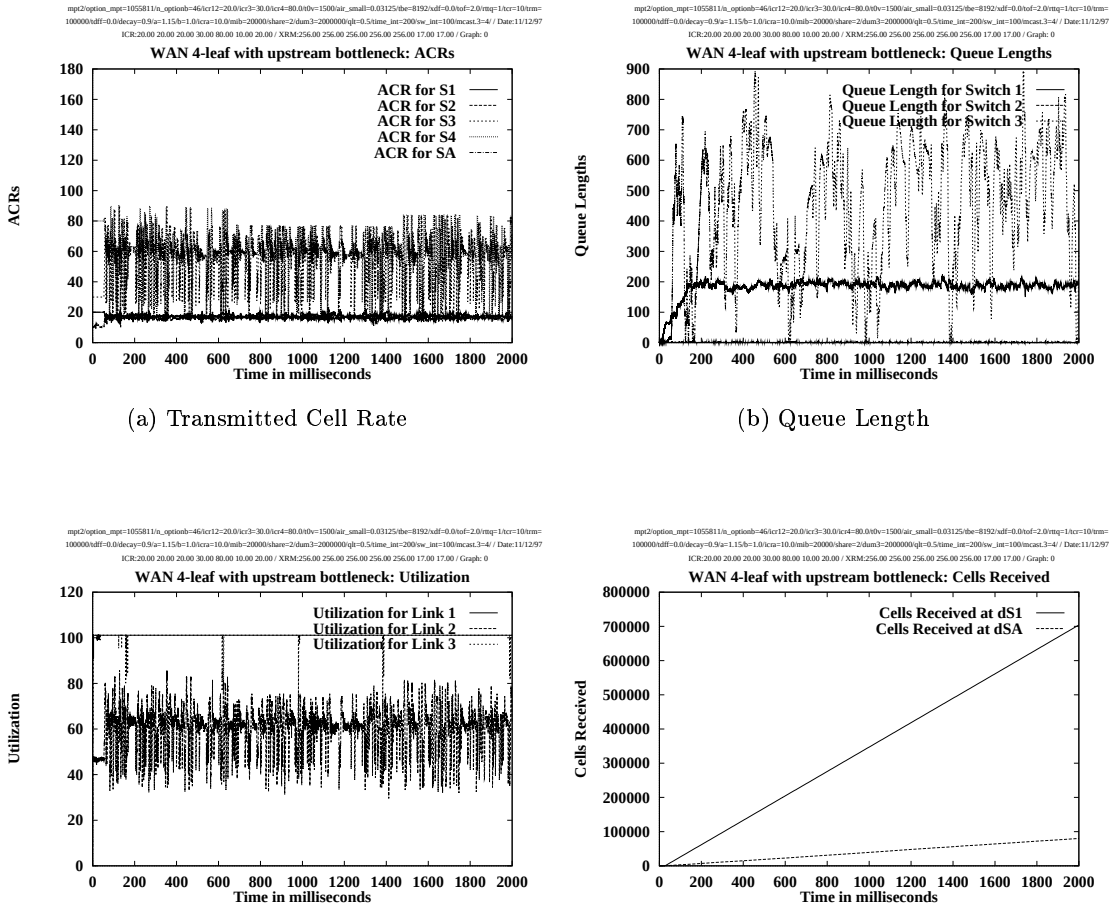
Source-based fairness divides bandwidth fairly among active sources, ignoring VC memberships. Source-based switch algorithms operating in VC merging switches need to avoid distinguishing among sources in the same VC. Key lessons learnt from this study include:

1. Source-level accounting should not be performed in multipoint rate allocation algorithms. For example, measuring the rates for each source, or distinguishing overloading and underloading sources cannot be performed. If such accounting is performed at the VC level or the flow level, an additional mechanism to divide VC or flow bandwidth among sources is necessary.

2. Estimating the effective number of active sources in order to divide the available capacity among them is very difficult in multipoint connections, since it is impossible to distinguish among senders in the same multipoint VC with VC merging implementations.

3. The only information a multipoint rate allocation algorithm can use is the information supplied in RM cells, in addition to *aggregate* measurements of load, capacity and queuing delays.

4. CCR values from BRM cells should not be used in computing rate allocations for sources in multipoint connections, since the CCR value can be that of another source that does not go through the switch performing the computation. That source may have a much higher bottleneck rate, and using its CCR can result in unfairness.



(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 16: Results for a WAN multipoint-to-point configuration with an upstream bottleneck (long LINK3, short interval)

5. CCR values from FRM cells can be used to compute rate allocations for sources in multipoint connections, even though the CCR used to compute the rate for a source may not actually be the CCR value of the source. This does not create problems due to the properties of the merged flow (see section 4 for more detailed explanation). The maximum CCR value seen in an interval can be used instead of the CCR of the source. Exponential averaging of the maximum CCR seen or maximum ER given can further improve the performance of the algorithm.

6. Merge point algorithms should avoid changing the BRM to FRM ratio at the sender or inside the network, to maintain the rate of feedback that the source requires, and avoid excessive overhead in the network. Scalability of the scheme is also affected by these ratios. Excessive complexity, noise, and response time can also be avoided by returning the BRM cells coming from the root, instead of turning around the RM cells at the merge points.

In this contribution, we have developed and simulated an algorithm for computing source-based fair allocations for multipoint-to-point and point-to-point connections. The algorithm uses simple aggregate measurements and maximum CCR values from FRM cells during successive intervals to perform rate computation. The algorithm exhibited very good behavior for the configurations tested.

More extensive performance analysis and convergence proofs are crucial to examine the fairness, complexity, overhead, transient response, delays, and scalability tradeoffs in multipoint algorithm design. Extending multipoint-to-point schemes for multipoint-to-multipoint connections can be performed by combining point-to-multipoint algorithms (such as those developed in [1]) with the multipoint-to-point algorithm.

References

[1] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shivkumar Kalyanaraman, Sastri Kota, and Pradeep Samudra. Feedback consolidation algorithms for ABR point-to-multipoint connections. ATM Forum/97-0615, July 1997. Extended version to appear in the Proceedings of IEEE INFOCOM 1998.

[2] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Sastri Kota, and Pradeep Samudra. Fairness for ABR multipoint-to-point connections. ATM Forum/97-0832, September 1997.

[3] The ATM Forum. The ATM forum traffic management specification version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[4] Matthias Grossglauser and K. K. Ramakrishnan. SEAM: Scalable and efficient ATM multipoint-to-multipoint multicasting. In *Proceedings of the IEEE INFOCOM*, April 1997.

[5] Juha Heinanen. Comments on the multipoint-to-point base line text. ATM Forum/97-0707, September 1997.

[6] Juha Heinanen. Multipoint-to-point VCs. ATM Forum/97-0261, April 1997.

[7] Jeffrey M. Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, COM-29(7):954–962, July 1981.

[8] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan. ERICA switch algorithm: A complete description. ATM Forum/96-1172, August 1996.

[9] Mark Jeffrey. Scope, concepts and issues for the new multiway BOF. ATM Forum/96-0628, June 1996.

[10] Wenge Ren, Kai-Yeung Siu, and Hiroshi Suzuki. Performance evaluation of multipoint-point ABR and UBR. ATM Forum/96-1402, October 1996.

[11] Wenge Ren, Kai-Yeung Siu, Hiroshi Suzuki, and Masayuki Shinihara. Multipoint-to-multipoint ABR service in ATM. *Submitted to IEEE Journal on Selected Areas in Communications*, 1997.

[12] A. Viswanathan, N. Feldman, Rick Boivie, and Rich Woundy. ARIS: Aggregate route-based IP switching. IETF Internet Draft: draft-viswanathan-aris-overview-00.txt, March 1997.

[13] Indra Widjaja, Steve Wright, and Amalendu Chatterjee. Interworking of VP-merge, VC-merge, and non-merge ATM switches in a multipoint-to-point environment. ATM Forum/97-0768, September 1997.

All our papers and ATM Forum contributions are available through http://www.cis.ohio-state.edu/~jain/