****

ATM Forum Document Number: ATM_Forum/97-0617

****

Title: Worst Case Buffer Requirements for TCP over ABR

****

Abstract:

We extend our earlier study on the buffer requirement problem for TCP over ABR. Here, a worst case sceanrio is generated such that TCP sources send a burst of data under the conditions when the sources have high congestion window the ACR (allowed cell rate) rate for ABR is high. We find that ABR using ERICA+ switch algorithm can maintain the queue lengths (hence the buffer requirements) bounded even for the worst case for a reasonable number of sources. We present analytical arguments for the expected queue length and simulation results for varying number of sources and parameter values.

****

Source:

Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal & Sonia Fahmy
The Ohio State University (and NASA)
Department of CIS

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

Pradeep Samudra
Director, System Engineering
Broadband Network Lab
Samsung Telecom America, Inc.
1130 E Arapaho
Richardson, TX 75081
Phone: (972) 761-7865
email: psamudra@telecom.sna.samsung.com

****

Date: July 1997

****

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

****

Notice:
This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.
****

1. Introduction:

We extend our earlier studies [1,2,7,8] of buffer requirements of TCP over ABR. In those studies, we had shown that to achieve zero loss ABR service requires switch buffering which is only a small multiple of round trip times of the feedback delay. The buffering depends heavily upon the switch scheme used.

One of the issues related to buffer sizing is that it is possible for a source to get a high ACR and keep it. As long as, it keeps sending a packet before 500 ms, the use-it or lose-it policy will not be triggered. The source can then use the high ACR to dump a large amount of data suddenly. The effect can be amplified with many sources trying to do the same. This contribution studies this particular case.

In this study we generate a worst case in which the TCP sources have a high congestion window and all ACR rates are high. Under such a condition the TCP sources are made to dump a huge burst of data into the network. We show that even under these circumstances ABR using a good switch algorithm like ERICA+ [3] performs well and controls the queues. In our previous simulations we had used only upto 15 TCP sources. In this study we present results of simulation which have upto 200 TCP sources.

2. Generating Worst Case TCP traffic

TCP has its own congestion control mechanism. It uses the "slow start" mechanism which is explained in detail in [4,5]. The amount of traffic which TCP can send into the network is dependent on TCP congestion window size and the ACR value.The maximum burst possible which TCP can send is equal to the maximum congestion window size.

The TCP takes log(congestion window) number of round trips to reach the maximum congestion window. In normal TCP activity when this maximum congestion window is achieved, the ACRs may not be high. In order to generate worst case conditions in which TCP has maximum congestion window size and the ACR is high, we designed the following sceanrio.

The configuration used is the N TCP Sources connected to N TCP Destinations via two switches and a bottleneck link (LINK1). The following figure shows the configuration used.

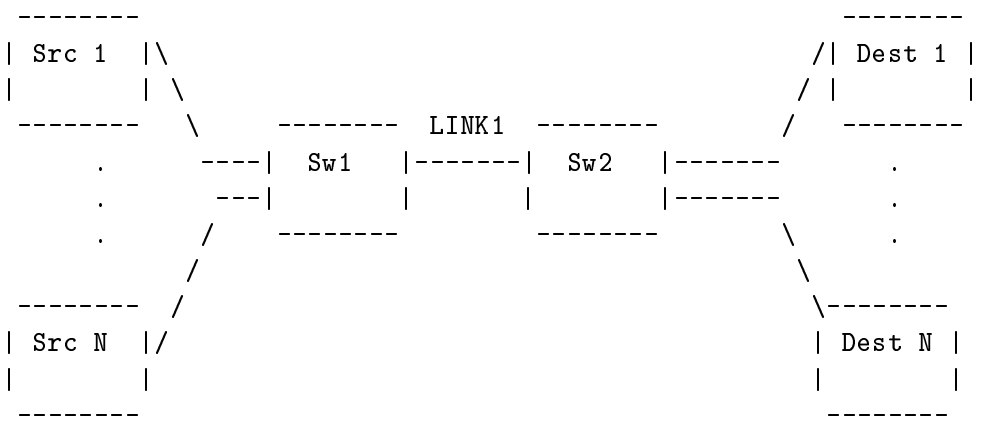


Figure 1: The "N Source - N Destination" Configuration

Initially to build up the congestion window each source sends one segment of data every 't' milliseconds. One thousand such segments are sent so that congestion window reaches a maximum for all the sources. The sources send segments in a staggered manner, so that not all sources send the segments simultaneoulsy. This is done so that the TCP data can be sent without overloading the network. Since the network is not overloaded, the ACRs are high when the congestion window reaches the maximum values. At this point all the sources synchronize and send a burst of data (burst size equal to maximum congestion window). Since every 't' milliseconds, one segment is sent and one thousand segments are sent by each source, the sychronization time is just after 1000*t milliseconds. This is the worst case burst size which arises due to the TCP sources.

3. TCP Options and ERICA+ parameters

We use a TCP maximum segment size (MSS) of 512 and 1024 bytes.

The MTU size used by IP is generally 9180 bytes, so there is no segmentation caused by IP. Since our simulations are performed under no loss conditions, the results hold even for TCP with fast retransmit and recovery or TCP with SACK (Selective Acknowledgement).

The TCP data is encapsulated over ATM as follows. First, a set of headers and trailers are added to every TCP segment. We have 20 bytes of TCP header, 20 bytes of IP header, 8 bytes for the RFC1577 LLC/SNAP encapsulation, and 8 bytes of AAL5 information, a total of 56 bytes. Hence, every MSS of 512 bytes becomes 568 bytes of payload for transmission over ATM. This payload with padding requires 12 ATM cells of 48 data bytes each.

The ERICA+ algorithm is an extension of ERICA which uses the queueing delay as an additional metric to calculate the feedback. ERICA+ uses four parameters : a target queueing delay (T0 = 500 microseconds), two curve parameters (a = 1.15 and b = 1.05), and a factor which limits the amount of ABR capacity allocated to drain the queues (QDLF = 0.5). In our previous study [6] we had enhanced our ERICA+ algorithm to handle variance in traffic. In our current simulations, this exponential averaging option is included to handle the variances in the traffic. The ERICA+ algorithm is designed to handle short spikes in ACR, achieve fairness and correctly count bursty sources.

4. Analytical predicition

In this section we derive the analytical value for the queue lengths. The queue length is given in two formulae for underloaded and overloaded networks.

```
Qlen = N * floor(cwnd_max/48)     for N <= floor(1000*t/g) (forumla 1)

     = N * 367 * t  cells         for N  > floor(1000*t/g) (forumla 2)
```

where

N	- number of sources
cwnd_max	- maximum congestion window size
t	- time between consecutive segments
g	- time between sources sending segments

Initially, for a small number of sources, the network is underloaded, so the ACRs are high when the burst occurs. When all the sources send the burst simultaneously, the whole burst can be sent into the network and this results in large switch queues.

Let 'cells_per_mss' be the number of cells required to send a segement and 'mss' be the maximum segment size.

The burst size due to one source = floor(cells_in_mss*(cwnd_max/mss)) cells

So if there are N sources, the expected queue length is

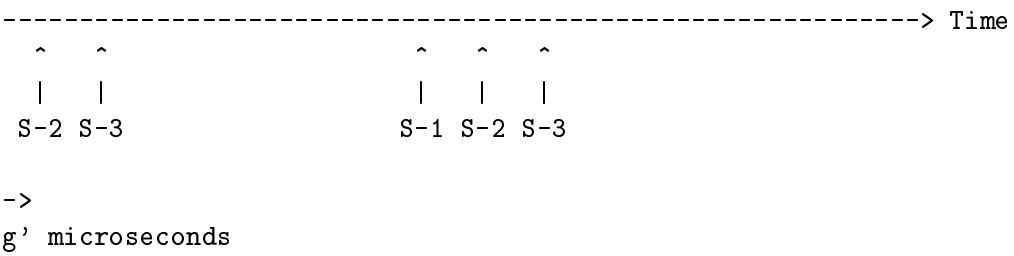
Qlen = N * floor(cells_in_mss*(cwnd_max/mss)) cells

Substituting cells_in_mss = ceiling(mss/48) in the above we get

Qlen = N * floor(cwnd_max/48) cells

Under no loss condition, the TCP congestion window increases exponentially in the "slow start" mechanism until it reaches the maximum value. As the number of sources increase the load of the network increases. In the simulation every source sends a segment of size 512 bytes every 't' milliseconds. The time between two sources sending the segment is 'g' microsecond. At time 0, source 1 sends a segment, then at time 'g' microsecond, source 2 sends, at time '2*g' microsecond source 3 sends and so on. Also source-1 sends its second segement at time 't' millisecond and so on (See figure 2).

Figure 2: Timing diagram: sources sending segments



Legend

S-<i> - Source number 'i'.

The network becomes overloaded when the input rate is greater than the output rate. In the simulation, the network will get overloaded if number of sources

```
 N >  (t ms / g microsecond)
   > 1000 * t/ g
   > floor(1000*t/g)
```

(For t = 1 millisecond, g = 50 microsecond, N = 1000*1/50 = 20)

Once the network is overloaded the ACRs are low and each of the N sources gets approximately (1/N)th of the bandwidth. Since the ACRs are low the burst which occurs after 1 second should not give rise to large queues. There are still switch queues which occur initially when the network has still not realized that it is overloaded.

The TCP congestion window grows exponentially whenever it receives an acknowledgement. Let 'd' be the length of the links in kms. The acknowledgement arrives after round trip time of 30*(d/1000) ms. But by this time the source would have generated more segements to send. After each round trip, each source will send twice the number of segments. A larger round trip can give rise to a larger burst. The networks get overloaded when the number of segments sent just fills up the pipe.

Let k be the number of round trips it takes to overload the network. Time taken for transmitting one cell is 2.72 microsecond. Hence, in 367*t. Each segement gives rise to 'cells_in_mss' cells. Therefore, the network gets overloaded when

```
        2^k* cells_in_mss = 367*t
=>      k = log (367*t/cells_in_mss)
               2
```

For t = 1 millisecond. mss = 512

```
=>      k = 4.934
=>      k ~= 5
```

(~= means approximately equal to)

Hence, in this case after 6 (= k+1) round trips the network detects the overload if N > 20. The maximum queues should occur after 6 round trips for N > 20.

The value of the maximum queue in an overloaded network can be derived as follows.

The burst due to one source = 2^k * cells_in_mss = 367*t cells

Hence the burst due to N sources = N*367*t cells

For t = 1 millisecond, g = 50 microsecond, mss = 512 and cwnd_max= 65536, we get

```
Qlen = N*1365                    for N <= 20
     = N*367                     for N > 20
```

5. Simulation results

We present the maximum queues at the bottleneck switch for the simulations where the number of sources (N) varies from 2 to 200. The results are shown in Table 1. The same information in graphical form will be presented at the ATM Forum and will be available at our web site after the Forum meeting.

The length of all the links are 1000 km, giving rise to 15 ms propagation delay from source to destination. The round trip time (RTT) is 30 millisecond. The feedback delay (the delay between the source and the bottleneck link) is 10 ms. A round trip of 30 ms corresponds to 11029 cells. The other parameter values are 't' = 1 millisecond, 'g' = 50 microsecond, 'mss' = 512 bytes, 'cwnd_max' = 65536.

The maximum queue length for sources N <= 20 was at time around 1 second. For N > 20, the maximum queue was at time around 300 millisecond (which agrees with the analytical prediction). The queue lengths increase with the number of sources as expected. The queue lengths also correspond to the burst size sent by the sources. But, this is true only when the number of sources are less than or equal to 20. For N = 30, there is sharp decrease the maximum queue length and again it starts (for N > 40) increasing but now it increases at a slower rate.

From the Table 1, it can be seen that for N <= 20, the simulation agrees well with the analytical values. For N > 30 initially the queue lengths in the simulation are higher than the analytical value, but later it becomes lower than the expected queue length. This can be explained as follows, initially the traffic generated in each cycle can be sent without pending queue in the next cycle. But as the number of sources increase (N > 80) in each

cycle there are some pending cells which get sent in the next cycles. Because of this the network detects the overload at lesser queue lengths than the analytical value. A more rigorous analytical derivation is necessary to take this into account.

Table 1: Effect of number of sources
Actual and Expected queue lengths

# TCP Sources	Max Queue size (cells)	Analytical Queue Size (cells)
2	1575	2730
3	3149	4095
5	6297	6825
10	14131	13650
20	29751	27300
30	20068	11010
40	19619	14680
50	24162	18350
60	28006	22020
70	30109	25690
80	31439	29360
90	34530	33030
100	38088	36700
110	43672	40370
120	44939	44040
130	44708	47710
140	44744	51380
150	46058	55050
160	48880	58720
170	50784	62390
180	49961	66060
190	53366	69730
200	55618	73400

To study the effect of different parameters a full factorial experiment was carried out for different parameter values. For a given number of sources the following parameters were varied: maximum segment size (mss = 512,1024 bytes), time between two sources sending segments (g = 50, 100 microseconds), time between successive segements of a source (t = 1, 10 milliseconds) and distance of the links (d = 1000,2000 kms).

Table 2: Full factorial experiment

#	mss/g/t/d	N=3	N=10	N=30	N=40	N=50	N=100
1	512/50/1/1000	3171	14273	20068	19619	24162	35687
2	512/50/1/2000	3171	14273	19906	27567	30872	75083
3	512/50/10/1000	3172	14274	45994	61854	77714	150453
4	512/50/10/2000	3172	14274	45994	61854	77714	150458
5	512/100/1/1000	3171	14273	19283	20080	24164	NA
6	512/100/1/2000	3171	14273	21241	32314	35961	NA
7	512/100/10/1000	3172	14274	45994	61854	77714	NA
8	512/100/10/2000	3172	14274	45994	61854	77714	NA
9	1024/50/1/1000	3040	13680	18650	18824	23542	NA
10	1024/50/1/2000	1542	5612	19131	22934	29163	NA
11	1024/50/10/1000	3040	13680	44080	59280	74480	NA
12	1024/50/10/2000	3041	13681	44081	59281	74481	NA
13	1024/100/1/1000	3040	13680	18591	19600	24314	NA
14	1024/100/1/2000	1403	5556	17471	24412	30533	NA
15	1024/100/10/1000	3040	13680	44080	59280	74480	NA
16	1024/100/10/2000	3041	13681	44081	59281	74481	NA

(NA - data not available)

Table 2 shows for number of source N=3,10,30,40,50,100 the maximum queue sizes for 16 experiments. From Table 2, the following observations can be made. These observations support the analytical results.

- In line numbers 10 and 14 for N = 3,10 the queue size is less compared to other lines, since in these cases congestion is detected early, so the ACRs are low when the burst is sent. Queue length is given by the formula 2.

- For N = 30,40,50,100, when t = 10 millisecond, the queue lengths agrees with one given by the formula 1.

- If the network is overloaded then a larger round trip time gives larger queue lengths. (e.g., see line 1 and 2 for N = 30,40,50).

- Lines 3 and 4 indicate that in underloaded conditions the queue length is given by the formula 1.

- As expected the segment size of 512 and 1024 do not affect the queue sizes.

6. Summary

We generated artifically a worst case scenario and studied the buffer requirement problem for TCP over ABR. The buffer size required depends heavily on the switch scheme used. Our simulations are based on our ERICA+ switch scheme. For worst case scenario generated both analytical prediction and simulation results for queue lengths (and hence buffer requirements) are given. The simulation agrees well with the analytical prediction. The queue lengths is affected by maximum congestion window size the round trip time, network congestion (overloaded or underloaded) and number of sources. It is not affected by maximum segment size.

7. References:

[1] R.Jain, S.Kalyanaraman, R. Goyal, S.Fahmy, S.M.Srinidhi, "Buffer Requirements for TCP over ABR," AF-TM 96-0517, April 1996.

[2] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal and Seong-Cheol Kim, ''Buffer Requirements For TCP/IP Over ABR,'' Proc. IEEE ATM'96 Workshop, San Francisco, August 23-24, 1996.

[3] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks, Part I: Description," submitted to IEEE Transactions on Networking, January 1997. Also presented as AF-TM 96-1172, August 1996.

[4] Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy, Fang Lu, Saragur Srinidhi, "TBE and TCP/IP traffic," ATM Forum/96-0177, February 1996.

[5] V. Jacobson, "Congestion Avoidance and Control," Proceedings of the SIGCOMM'88 Symposium, pp. 314-32, August 1988.

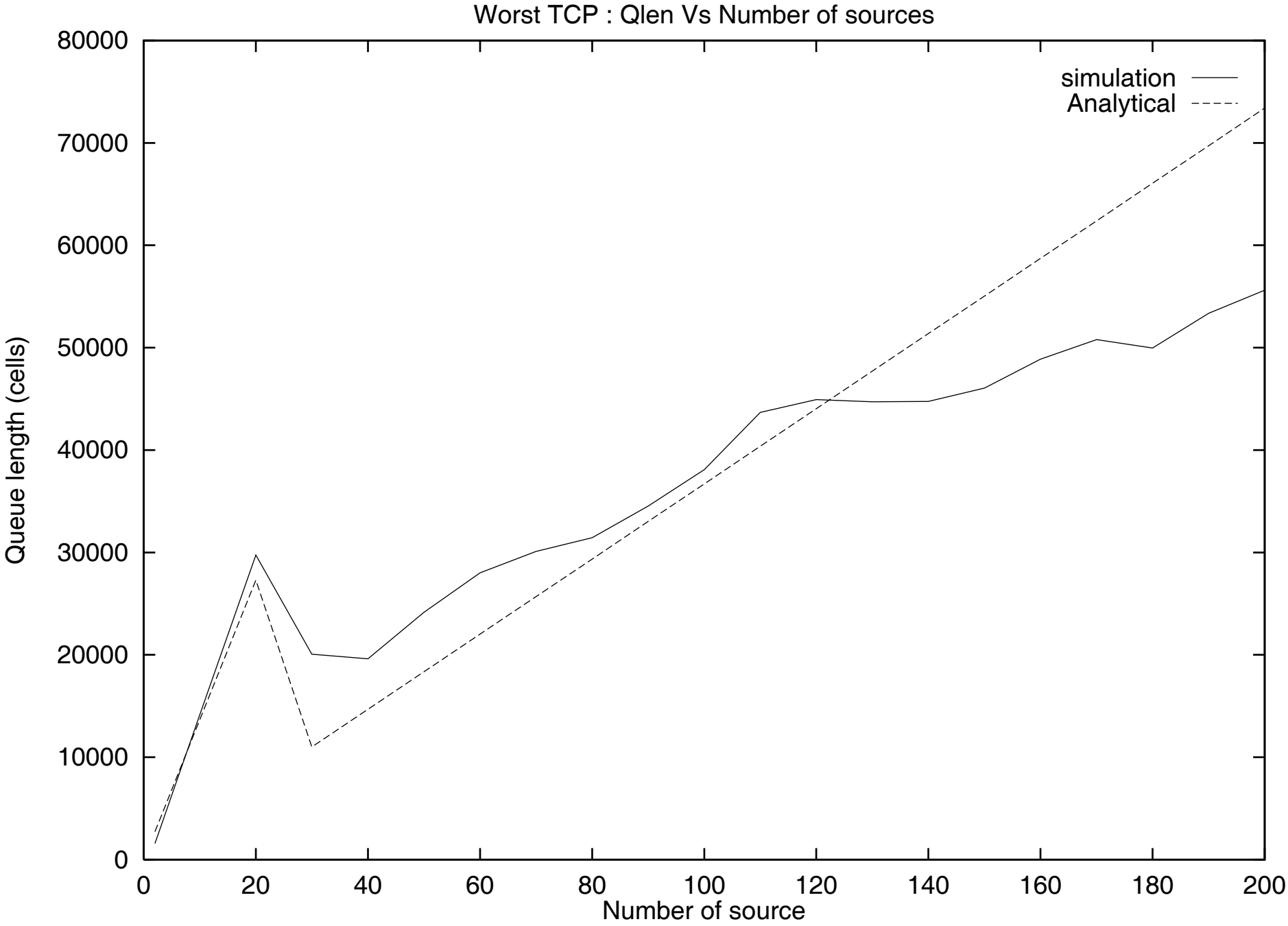
[6] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, Jianping Jiang, Seong-Cheol Kim, "Performance of TCP over ABR on ATM backbone and with various VBR traffic patterns," ATM Forum/96-1294, October 1996.

[7] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal and Seong-Cheol Kim, ''Performance and Buffering Requirements of Internet Protocols over ATM ABR and UBR Services,'' Submitted to IEEE

Communications Magazine, January 1997 issue.

[8] Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy and Seong-Cheol Kim, ‘‘Performance of TCP/IP Using ATM ABR and UBR Services over Satellite Networks,’’ IEEE Communication Society Workshop on Computer-Aided Modeling, Analysis and Design of Communication Liks and Networks, Mclean, VA, October 20, 1996.

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


Worst TCP : Qlen Vs Number of sources
simulation
Analytical
Queue length (cells)
Number of source
0
10000
20000
30000
40000
50000
60000
70000
80000
0
20
40
60
80
100
120
140
160
180
200