Technical Report 2003-02-03
Internetworking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

Performance Analysis of TCP Interactive based World Wide Video Streaming over ABone: Jitter and Delay Management

Javed I. Khan and Raid Y. Zaghal
Networking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
233 MSB, Kent, OH 44242
javed|rzaghal@cs.kent.edu

Abstract—Interactivity in transport protocol can greatly benefit transport friendly applications. We have recently implemented an interactive version of TCP. The implementation has two components-- an interactive transport protocol over FreeBSD called iTCP and, a novel symbiotic MPEG-2 full logic transcoder, which can dynamically change video characteristics based on interactive congestion response inside network layer. We have experimented with the real system on the Active Network (ABone) using selected nodes in the U.S. and Europe. In this report we present the jitter and delay experiments of the live video streaming results to these sites. A second report contains the detail results from application level video quality experiments.

1. Introduction

Congestion control for time-sensitive multimedia traffic has remained a difficult problem. Most of the mechanisms for congestion control those have been proposed to date are based on delaying traffic at various network points. The more classical schemes depend on numerous variants of packet dropping in network, prioritization (graceful delay in router buffer) admission control (delaying at network egress points), etc. However, a key aspect of a vast majority of these schemes is that they introduce **time distortion** in the transport pathway of applications. Though time distortion does no harm to time insensitive traffic such as email forwarding or ftp data, but they work completely against the application if the traffic is time sensitive such as multimedia streaming or control data.

For last few years it has been felt that for multimedia applications, the applications themselves have to be more integrated in the solution. Particularly promising are the research in the new TCP friendly paradigm [KeWi00, ReHE00, SiWo98, PrCN00]. [SiWo98] presented a TCP rate-based pacing mechanism that particularly takes note of document transfer characteristics. [ReHE00] discussed a general framework where applications can control rates based on their end-to-end measurements (similar end-to-end technique is used in RealPlayer). There are also fully application level proposals. Due to the lack of convenient means to obtain network states several works suggested [BrGM99, Wolf97] sending multilevel redundant information for video. Also several other works investigated combining application specific information from several streams into one clearinghouse architectures for aggregated congestion control. For example, Congestion Manager [ABCS00, BaRS99] is a system layer component. It provisions aggregated congestion control when multiple streams from the same end-point attempt to send via a separate program called Congestion Manager (CM). [SiWo98] proposed building TCP friendly application where application relies on real-time transport protocol (RTP) mediated end-to-end measurement. CM tries to minimize congestion by coordination between multiple sending streams. [PrCN00] used multiple probing mechanics for aggregate congestion control.

There has been several promising work on network or system level issues to increase TCP friendly-ness. Though, the paradigm of 'friendly applications' almost by definition shifts a major part of the congestion management responsibility to the applications, interestingly relatively very few work exists that seriously looked into the corresponding issues that arise in an actual time-sensitive application while taking advantage of the suggested 'friendliness'. The dynamics of the two systems can lead to stability issues. Time sensitive applications themselves have substantial complexity in adapting. Rate adaptation for any advanced multimedia application in general is quite complex. It requires sophisticated layer 4+ techniques. It is highly unlikely that multimedia rate adaptations for any performance coding schemes (such as MPEG) can be performed at predominantly network or system layer.

It seems that an alternate strategy for time sensitive multimedia traffic should be the multimedia knowledge enriched rate control, which can work in symbiosis with the network condition. We have recently implemented an MPEG-2 ISO-13818-2 [ISO96, KYGP01, KhYa01, KhGu1] video streaming system, and a novel interactive version of TCP called TCP Interactive. The general principle we follow is simple and intuitive. It seems an effective delay conformant solution for time sensitive traffic may be built if the original data volume can be reduced by its originator-- the application[1].

However, a key element in any such scheme is that the application must be notified. Unfortunately, today's transport protocols do not support any interactivity with applications. It seems such non-interactivity has been inherited from the early days of networking interface research, when the applications were simple and did not require sophistication. In this report we will show that transport interactivity can bring major benefit to high performance and demanding applications. The particular scheme we propose here has the following novel aspects compared to other recent works:

- First, we suggest an active and direct notification mechanism by the underlying transport protocol, rather than using indirect end-to-end feedback tools. If there is any congestion, we propose an interactive transport protocol, which can directly notify the application.
- To demonstrate the efficacy of the principle, we have designed a corresponding video rate transcoder system that works in symbiosis with the network. This transcoder actively participates in a custom symbiotic *exponential-back-off and additive-increase* like scheme in application layer with deep application level knowledge. (This is also one of the first to our knowledge) resulting in much more effective joint quality/delay sensitive communication.
- The resulting scheme is similar in spirit to the TCP friendly approaches. However, there is a fundamental difference in how it is done. We expect network (or system) layers to remain as simple as possible. The means and techniques for rate reduction remain with the producer application. The responsibility of the network layer is simply to pass on only selected end-point events to the applications.

As, we will show the scheme is not only intuitive and simple, but also surprisingly effective compared to many other recently proposed schemes, which involve much more complex system/network layer reorganization.

The result presented in this report is not simulation; rather report from a real implementation of the concept system that we have completed very recently. The implementation has two components-- an interactive transport protocol over FreeBSD that we called iTCP and, a novel symbiotic MPEG-2 full logic transcoder [KYGP01, KhYa01], which is capable of working in tandem with the interactive transport. The transcoding model has been developed by closely following the MPEG-2 Test Model 5 (TM5). MPEG-2 TM-5 signifies a real video coder with substantial complexity of itself. While the detail can be found in [Mpeg00], we describe the salient part of the rate control architecture that is critical to this symbiosis in [KhGR02].

In this report we only present the delay and jitter performance of the scheme on the (ABone) Active Network as we experimented the system on several ABone nodes in the U.S. and Europe. To learn more about the event model and the API of the iTCP, check our technical report [KhZ03a]. Also additional experiments results on application layer performance is in the technical report [KhZ03c].

2. The ABone Active Networks

An important feature of our experiment is using a real implementation of the proposed transport protocol (iTCP) and the MPEG-2 transcoder. Furthermore, we wanted to run the experiment on the real Internet environment. To conduct our experiment we wanted to run our video player on a number of remote hosts around the world and measure performance in each case. We could have done this by "*telneting*" to those remote nodes. But this would have required preparation and communication with people around the world to setup accounts and administer them. Furthermore, this will not be flexible nor practical if we decide to switch to a new set of remote nodes. Therefore, we decided to use the ABone Active Networks.

In Active networks, the routers or switches of the network can perform customized computations on the messages flowing through them. These networks are

[1] It is interesting to note, that the idea of application and network symbiosis have been mentioned for quite some time. However, almost no study exists which has focused on it.

active in the sense that nodes can perform computations on the contents of the packet. As far as we are concerned, we wanted to be able to run our video player on a selected set of ABone nodes and measure the performance of the video session. In that respect, the ABone provided a convenient testbed for us to run the experiment. We simply sent a modified version of our video player to the ABone administrator at the ABone Coordination Center (ABOCC) to be placed on the trusted code server at (http://bro.isi.edu/KENT). Then we configured and registered our iTCP machine (kawai.medianet.kent.edu) as a primary node on the ABone.

In addition to the iTCP machine we have 10 registered ABone nodes at Kent State University (mk00- mk09.maunakea.medianet.kent.edu). Four of these nodes run on FreeBSD and the rest run on Linux. At the time of our experiment (Feb. 2003) there were 24 Linux nodes, 5 Solaris nodes, and 12 FreeBSD nodes registered on the ABone. Since our player was compiled on Linux, we could use Linux nodes only.

3. Experiment Setup

This experiment describes the performance for the case of a MPEG-2 ISO/IEC13818-2 (176x120) resolution video encoded with base frame rate of 2 Mbps at main profile on the symbiotic transcoder.

Figure-1 illustrates the experiment setup. The video server runs on a classic TCP machine (*manoa*) and feeds the video stream into the transcoder, which runs on the iTCP machine (*kawai*). This machine is registered as a primary node on the Active Network (ABone). We also have ten other machines (*mk00 – mk09*) that are registered on the ABone as well. We used this cluster of ABone nodes to generate background cross traffic while the video is playing. We run the player on a selected remote ABone node using the Anetd LOAD command from (*kawai*). We repeated the experiment on five ABone nodes, three in the US and two in Europe. All five nodes are shown in Table-1.

In all runs, the transcoder subscribes with iTCP for two events: REXMT (retransmit timer out event) and DUPACK (third duplicate acknowledgment event) see Table-2. Also, we always turn on the event notification property of the iTCP. The only controlled parameter that we changed was the reduction property of the signal handler. When the reduction flag was set (*symbiosis=on*), the signal handler invokes the event handler to reduce the bit rate of the decoder. Otherwise (when *symbiosis=off*), the signal handler just records the event type and time in a log file.

We repeated the experiment ten times with each remote ABone node from Table-1, five times in the (*symbiosis=on*) mode and five times in the

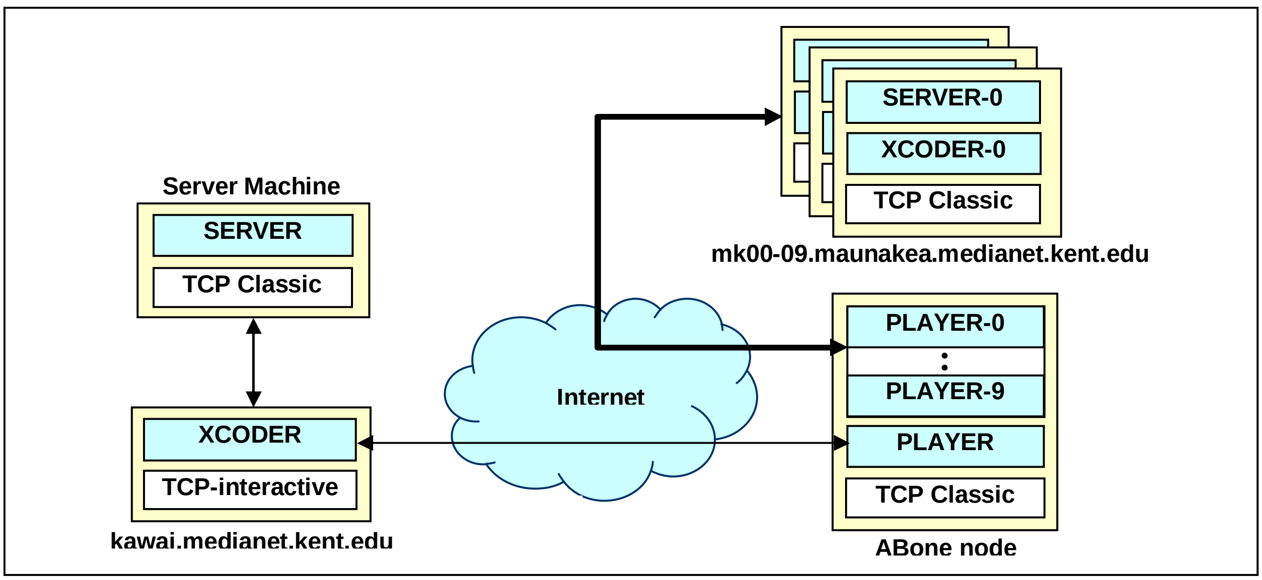

Figure-1. Experiment setup. The transcoder runs on the iTCP machine and the player runs on a remote ABone node. The mk00-09 cluster generates background cross traffic.

ABone node	IP	Country	Number Of Hubs	RTT (ms)		
				Avg	min	max
abone.fokus.gmd.de	193.175.135.49	Denmark	21	144	131	216
galileo.cere.pa.cnr.it	147.163.3.12	Italy	20	287	266	339
abone7.cs.columbia.edu	128.59.22.217	NY, USA	15	41	39	60
abone-01.cs.princeton.edu	128.112.152.62	NJ, USA	15	51	46	69
dad.isi.edu	128.9.160.202	CA, USA	16	65	65	68

Table-1. ABone nodes used to run the player in the experiment.

(*symbiosis=off*) mode. In each run we recorded two log files, one on the transcoder side (*kawai*), and one on the player side (the remote ABone machine). We retrieved the latter log file using the Anetd GET command. The transcoder recorded the following information for each frame in the video stream: *frame number, departure time, target bits, actual bits, and SNR values for Y, U, and V blocks.* Also, when an event signal is received from the iTCP, the signal handler records its type and timing. On the player side, the log file only records the arrival time of each frame.

In the following discussion we will regard the (*symbiosis=on*) mode to resemble iTCP and the (*symbiosis=off*) to resemble classic TCP. We made this resemblance since the (*symbiosis=off*) mode adds only the event notification property to the TCP. This small overhead is irrelevant and can be ignored in the overall system performance analysis.

4. Events and Timing Information

Table-3 shows several parameters to measure end-to-end performance at both the application and network levels on the five target ABone nodes. Part (a) of the table represents the results for the iTCP mode were symbiosis was applied and part (b) represents the results for the TCP classic mode. Each value in the table is an average of five runs on the specified ABone node. We show six parameters in Table-3: *average number of events, average referential jitter per frame, average inter-arrival time per frame, average flight time per frame, average time to transmit/play the entire video (1000 frames), and average frames per second.*

Subscribe flag (iTCP) = on
Event reception flag (EVENT) = on
Rate reduction flag (SYMBIOSIS) = on/off
Reduction Factor (ALPHA) = 0.55
Subscribed events = REXMT | DUPACK
Frame size = 176 x 120
Number of Frames = 1000

Table-2. Experiment and video parameters. Only the reduction flag (SYMBIOSIS) was changed in different runs.

To facilitate comparison between the two cases, we converted each parameter from the Table-3 into a bar graph. We show these bar graphs in Figure-2. In each bar graph the x-axis represents the five target ABone nodes and the y-axis represents the measured parameter. First, we notice that the average number of congestion events for both TCP and iTCP modes on all ABone nodes were relatively close (1.4 on iTCP vs. 1.6 on TCP). This observation justifies the comparison and enables us to make the assumption that both modes were running under similar network conditions. Direct observation of these bar graphs reveals the advantage of the iTCP mode over TCP mode.

4.1. Referential Jitter and Inter-Arrival Time

The '*Average Referential Jitter*' graph of Figure-2 shows that iTCP (or *symbiosis=on* mode) managed to substantially reduce the jittery behavior in all cases especially for the two nodes in Europe. To measure this quantity we found the difference between the actual arrival time of each video frame and the expected arrival time of the same frame in the ideal case. Those frames that arrive late will cause the jittery behavior of the video playback. This improvement was achieved because the iTCP managed to contain the buffer buildup and thus speedup the delivery of the frames that suffered from congestion.

The next graph in Figure-2 shows the '*Average Inter-Arrival Time*' per frame, which also represents the jitter. It is obvious that iTCP reduced that time per frame on all five nodes.

4.2. Video Timing and Frame Rate

The '*Average Flight Time per Frame*' graph shows the average time needed to transfer each frame from the transcoder to the player. Here we see a mixed behavior; while with '*italy*' and '*us-columbia*' nodes the iTCP outperformed TCP-classic, it lost for TCP with the rest of the nodes. From this observation we can predict that our interactive scheme cannot influence the flight time of the frame. This parameter was probably affected by the network load and the availability of bandwidth during each individual video session.

Next graph shows the '*Average Time per Video*'. Here we show the total time needed to transmit and play the video clip averaged for five runs per node. We think this parameter is important since it shows that iTCP managed to considerably reduce the overall delay of the video session even during severe congestion (e.g. with the '*Italy*' node there were about 50 seconds in favor of the iTCP mode). The last graph of Figure-2, '*Average Frames per Second*', shows the average frame rate of the video session on each node. This parameter is a direct consequence of the previous parameter (i.e. '*Average Time per Video*') and was calculated by dividing the number of frames in the video clip (1000 in our case) by the '*Average Time per Video*'.

5. MPEG-2 Frame Transport Efficiency

5.1. Frame Arrival Delay

Now we show the impact of TCP interactivity on frame arrival delay at the remote player. We took frame wise detail event trace of what happens to the first 1000 frames of the video at both sending and receiving ends. For a given discard threshold time in the receiving end we also traced which frame was successfully received or not at the receiving end of the MPEG-2 player. In Figure-3 we plot the delay experienced by the video frames. Each part [(a)-(e)] of Figure-3 plots the frame arrival time for one of the target Abone nodes. In each case we show five runs where the symbiosis property of the transcoder is activated (marked as R1-R5, *symbiosis=on*) and five other runs where the symbiosis is turned off (marked as R1-R5, *symbiosis=off*). In the figure we plot the (*symbiosis=on*) or iTCP runs in the shades of red and the (*symbiosis=off*) or TCP-classic runs in the shades of blue.

As it can be shown, after each congestion burst, TCP-classic continuously fell behind. The delay built up and hardly it could recover. This is shown by the step jumps in the delay line. The iTCP also suffered some step buildup, but it in most cases it was much smaller and it could recover after few seconds. Furthermore, in many runs of the TCP-classic mode, the buildup can be seen as a change in the slope of the line immediately after the step jump (e.g. R2 on '*focus*' and R5 on '*galileo*'). In the iTCP runs the line always followed the expected trend after the step jump.

5.2. Referential Jitter

In Figure-4 we plotted the jitter experienced by the frames in the five nodes. We took the difference between the expected ideal arrival time and the actual arrival time for each frame. A negative jitter means the frame arrived earlier than expected. Like the previous plot, the iTCP runs are shown in the shades of red while the TCP runs are shown in the shades of blue. Two things can be noticed in these plots; first, the step jumps in the iTCP runs were generally smaller than those in TCP-classic runs since the frames that faced congestion suffered less delay. This can be clearly seen in plot (A), (C), and (D) of Figure-4. Second, in some cases of the TCP runs, the jittery behavior of the video stream increased immediately after the huge step jump. This can be seen in the plot when the line moves upwards after the step jump and remains in that direction for 50-100 frames period. This behavior is clearly expressed in plot (C), (D), and (E). On the other hand, in all iTCP runs, the line either stayed horizontal, i.e. no change in the jitter after the step jump, or went down as a result of lower jitter.

As shown the iTCP drastically reduced the jittery behavior.

6. Conclusions and Current Work

In this report, we have presented a case of rate symbiosis mechanism in line with current advances in TCP friendly systems. We have presented the case through a simple 'interactive' generalization of the classical transport control protocol, and a novel implementation of a symbiotic MPEG-2 transcoder. We collected the results of our experiment by running the video session on the global Active Network (ABone) testbed.

In previous discussion we have demonstrated the case of quality conformant congestion control for time-sensitive video traffic. The approach exposed the overall advantage of network '*friendly*' applications. However, it also departs significantly from the mainstream TCP friendly systems that have been suggested recently in two senses; First, it does not add any new major component in network software structure. One of the principal strength of the proposed scheme is its relative simplicity at network layers –yet its effectiveness. It only expects some form of interactivity directly from the concerned network protocols as a general interface feature. Thus, there is no expectation of (or conflict with) additional services (such as combined congestion control from multiple applications).

Secondly, the applications do not have to be designed dependent on other auxiliary indirect probing tools or network utilities, nor it excludes their use when available. Some of the information measured by the auxiliary tools suggested by other approaches might be already available (or are being estimated/tracked) at lower layers anyway. At least this is the case with TCP congestion. The direct protocol interactivity we propose thus seems to be the logical path that can avoid potential duplication of efforts.

Nevertheless, the approach adds lesser but yet some complexity in the network layer. The augmentation of the notification feature increases the normal mode delay of TCP even it is slight. The actual cost depends on the intensity of coupling. Designer of application symbiosis unit must be aware of the potential cost of tight coupling between handler and caller. However, as shown by the results-- with a prudent design the impact on the network level transfer rate (based on low layer measurement), if any, can be widely surpassed by the gain made at application layer. However, an interesting safeguard of this scheme is that a wrong design will only affect the application at fault and will have no effect on network or others. However, the proposed interactivity is not an alternate to other network level schemes, rather is a complimentary scheme.

ABone node	Num. Of Events	Ref. Jitter	IAT per frame	Flight per frame	Time per video	Frames per Second
fokus (denmark)	2	7.1494305	0.115616	0.34110329	115.615629	8.70646714
galileo (italy)	1.2	7.46529	0.107585	0.27807673	107.585495	9.34654553
columbia (usa)	1.6	1.0135707	0.101358	0.19717850	101.357553	9.90596598
princeton (usa)	1.2	3.5122045	0.110104	0.22855333	110.164151	9.22516667
isi (usa)	1	5.604205	0.107402	0.21521991	107.421659	9.37435276
AVERAGE	**1.4**	**4.948940**	**0.108413**	**0.2520263**	**108.42889**	**9.3116996**

(A)

ABone node	Num. Of Events	Ref. Jitter	IAT per frame	Flight per frame	Time per video	Frames per Second
fokus (denmark)	1	18.284166	0.14174	0.334396874	141.7599768	7.903423414
galileo (italy)	2	36.890137	0.157235	0.357548888	157.2548856	6.554845167
columbia (usa)	1.6	7.0201728	0.112287	0.173041292	112.2868738	8.964446532
princeton (usa)	1.4	6.6170372	0.112419	0.230792746	112.419281	8.978581526
isi (usa)	2.2	10.950218	0.122707	0.300137613	122.7066482	8.238424173
AVERAGE	**1.64**	**15.95234**	**0.129278**	**0.2791834**	**129.28553**	**8.1279441**

(B)

Table-3. Event and timing information. Table (A) shows the results of the iTCP runs, while table (B) shows the results of the classic TCP-classic runs.

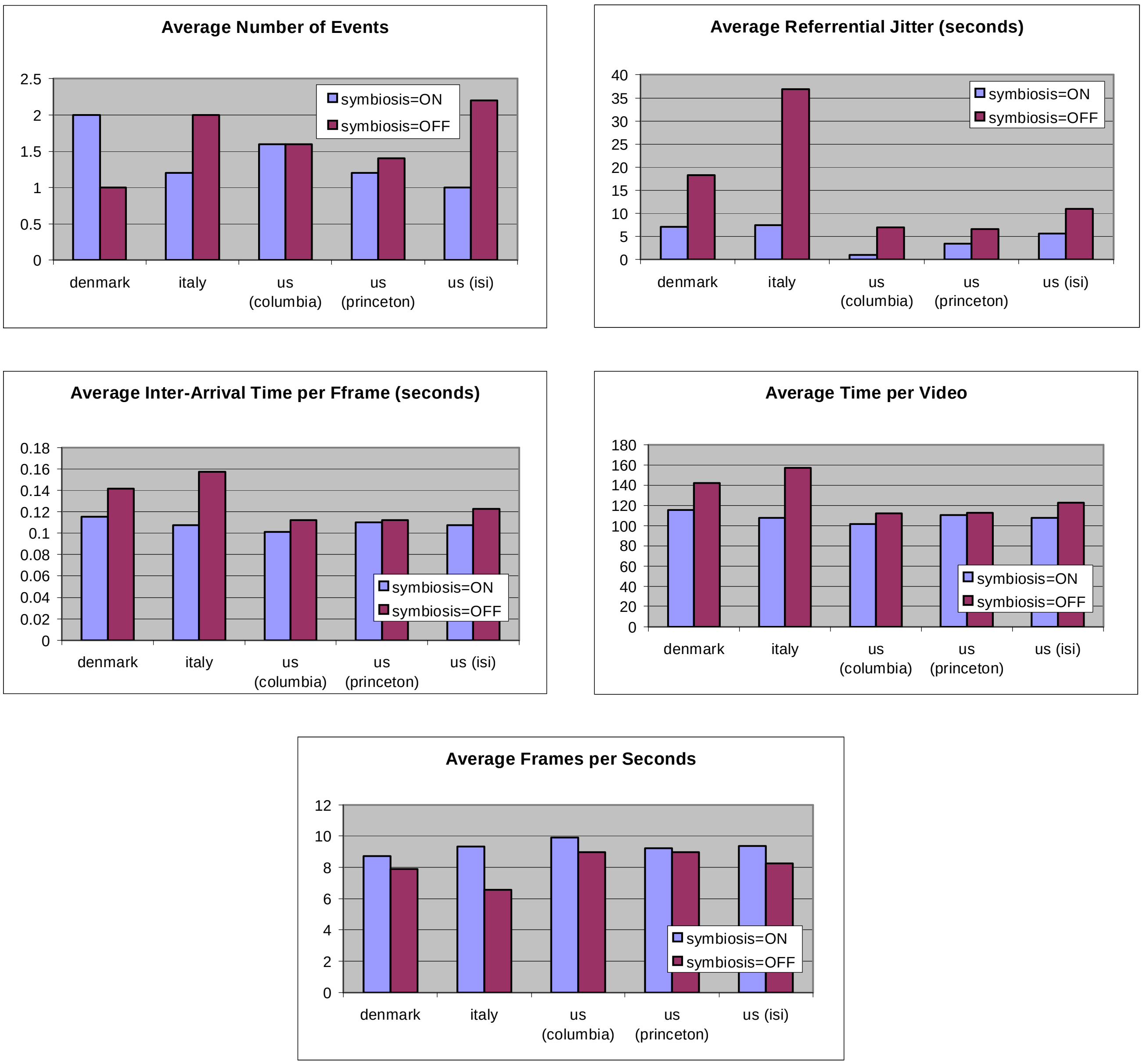

Figure-2. Events and timing information from Table-4. Each parameter is shown as a separate bar graph to facilitate comparison between the two modes of experiment (sym=ON|OFF).

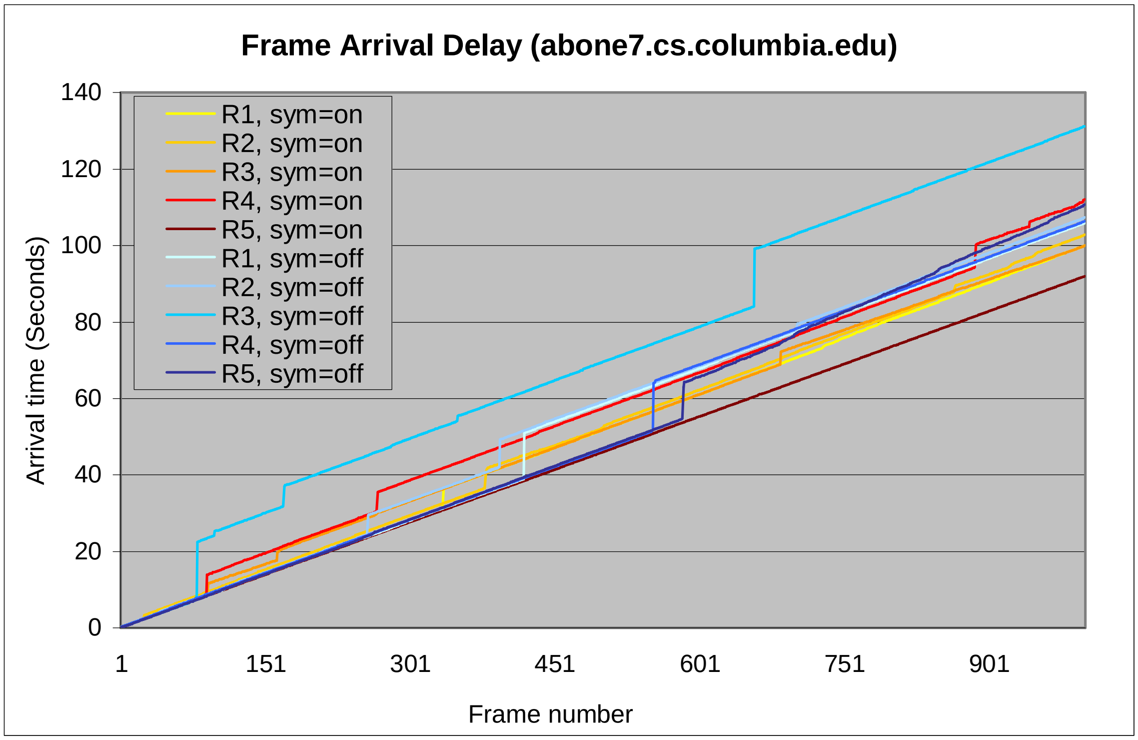

Figure-3 (A)

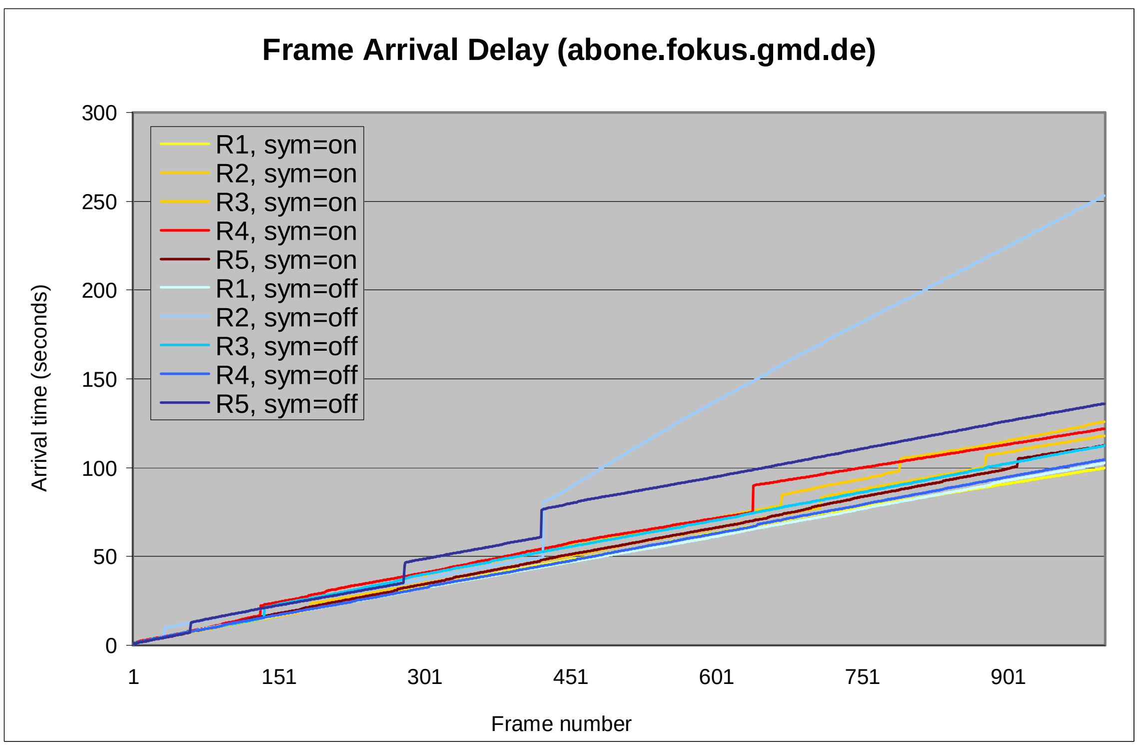

Figure-3 (B)

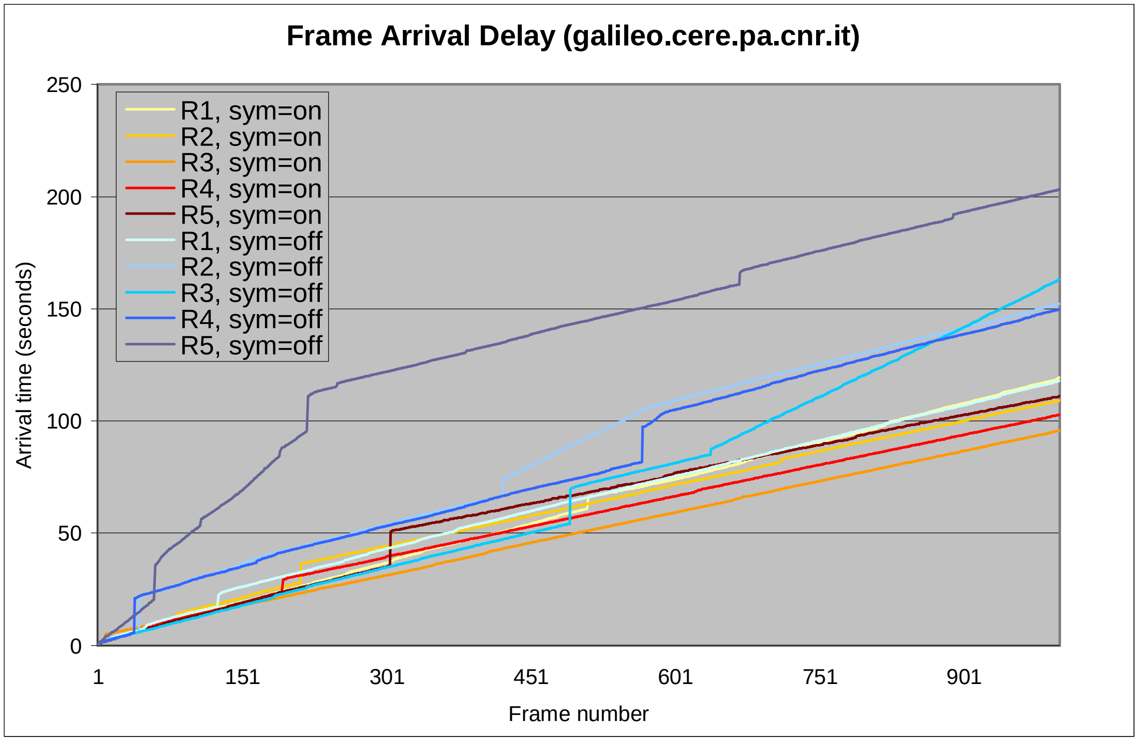

Figure-3 (C)

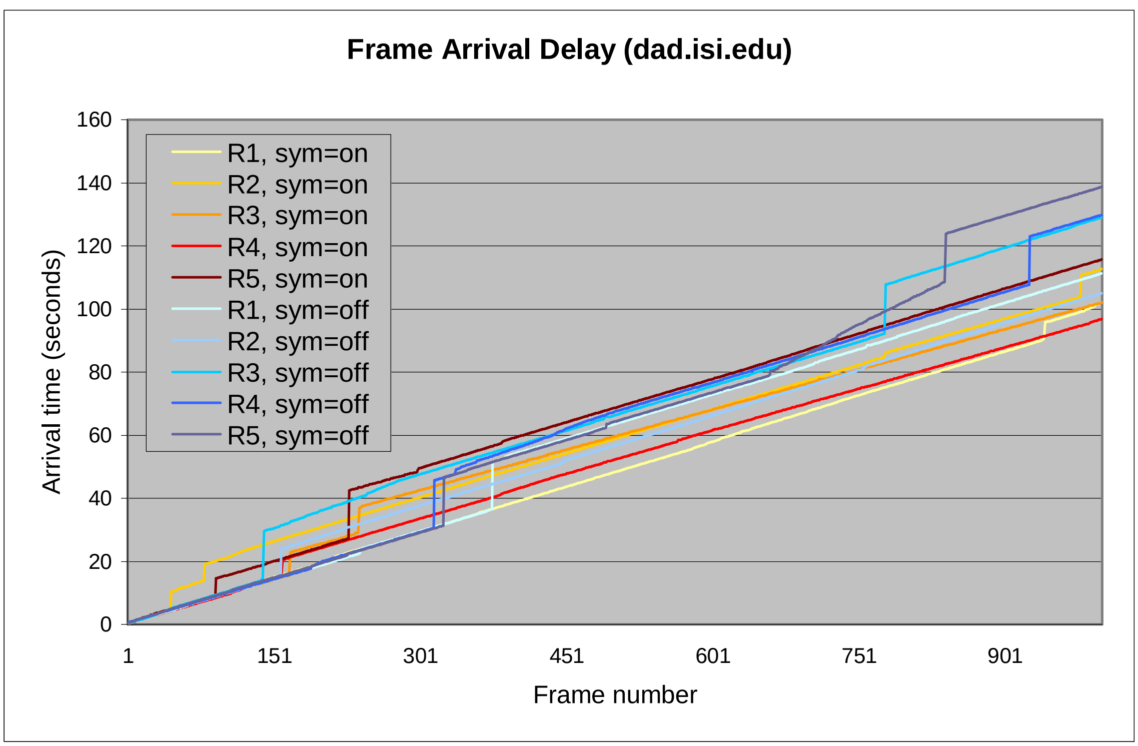

Figure-3 (D)

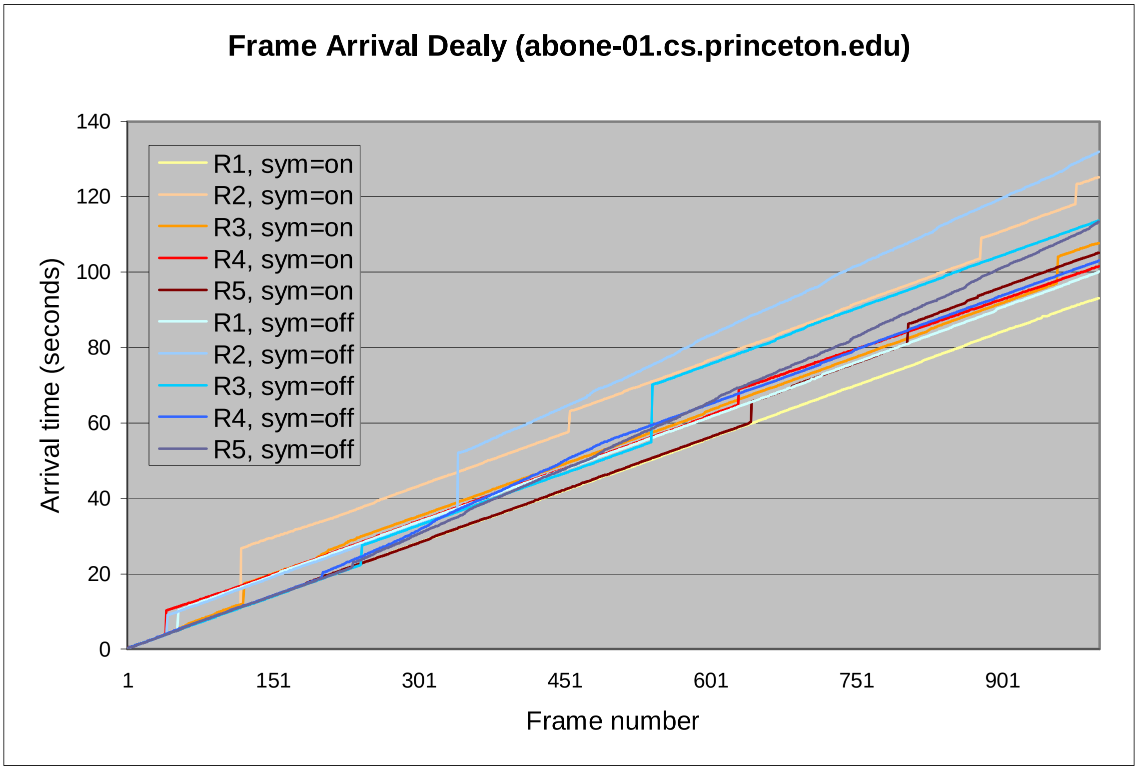

Figure-3 (E)

Figure-3. The arrival time of video frames. With each timeout event the backlog increases and can be observed as step jumps in the delay. The iTCP helps in gradually reducing these step jumps for consecutive loss events.

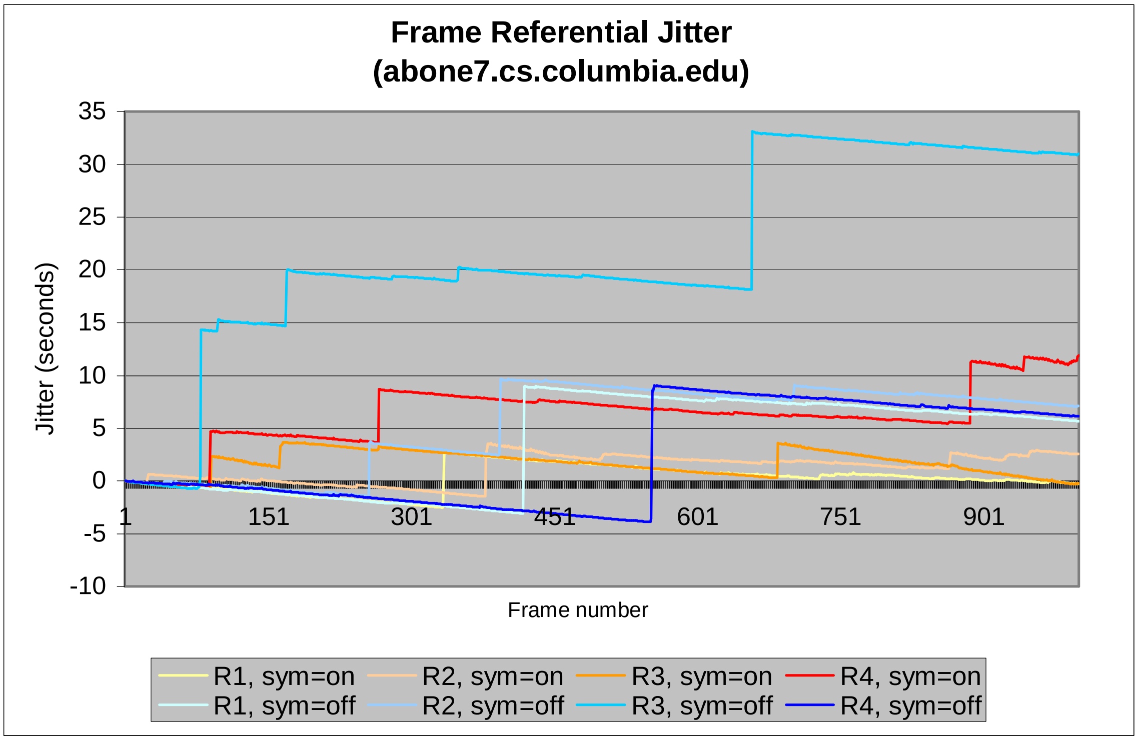

Figure-4 (A)

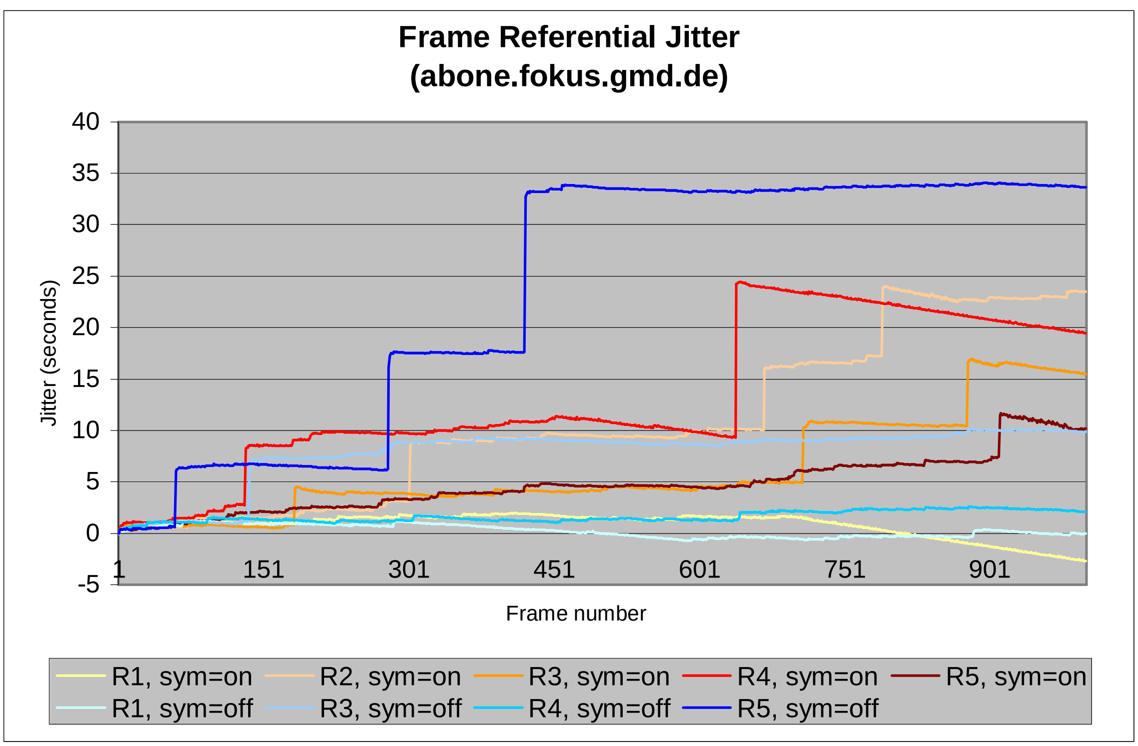

Figure-4 (B)

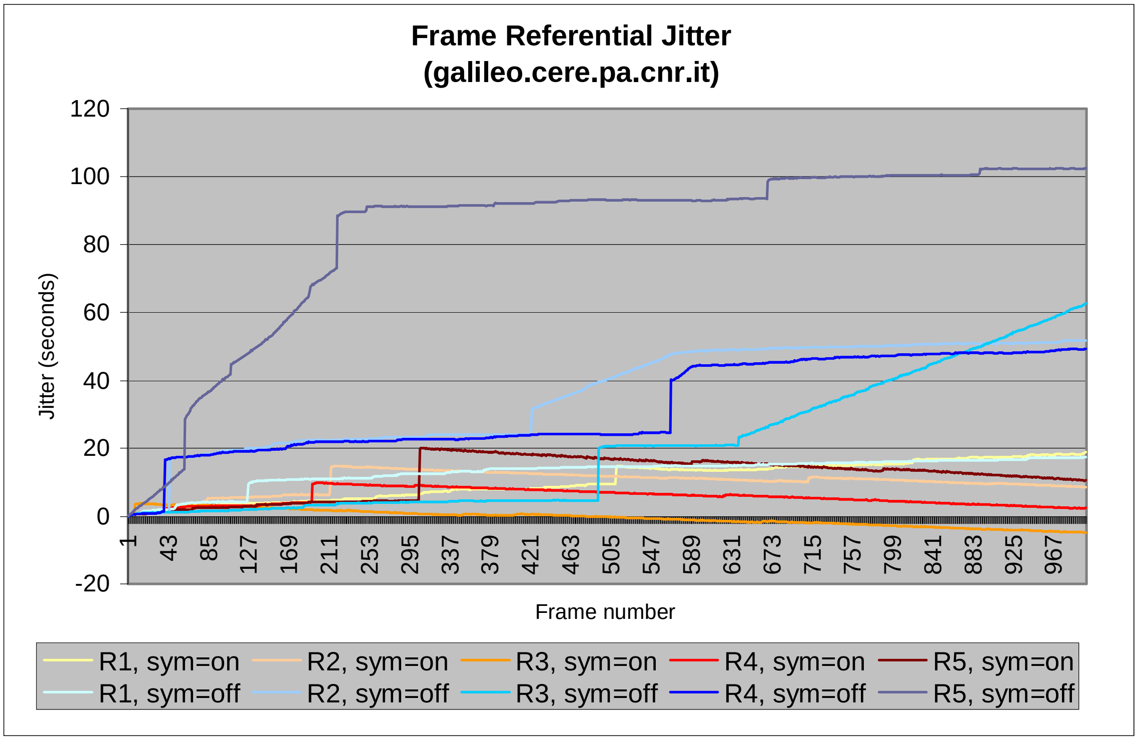

Figure-4 (C)

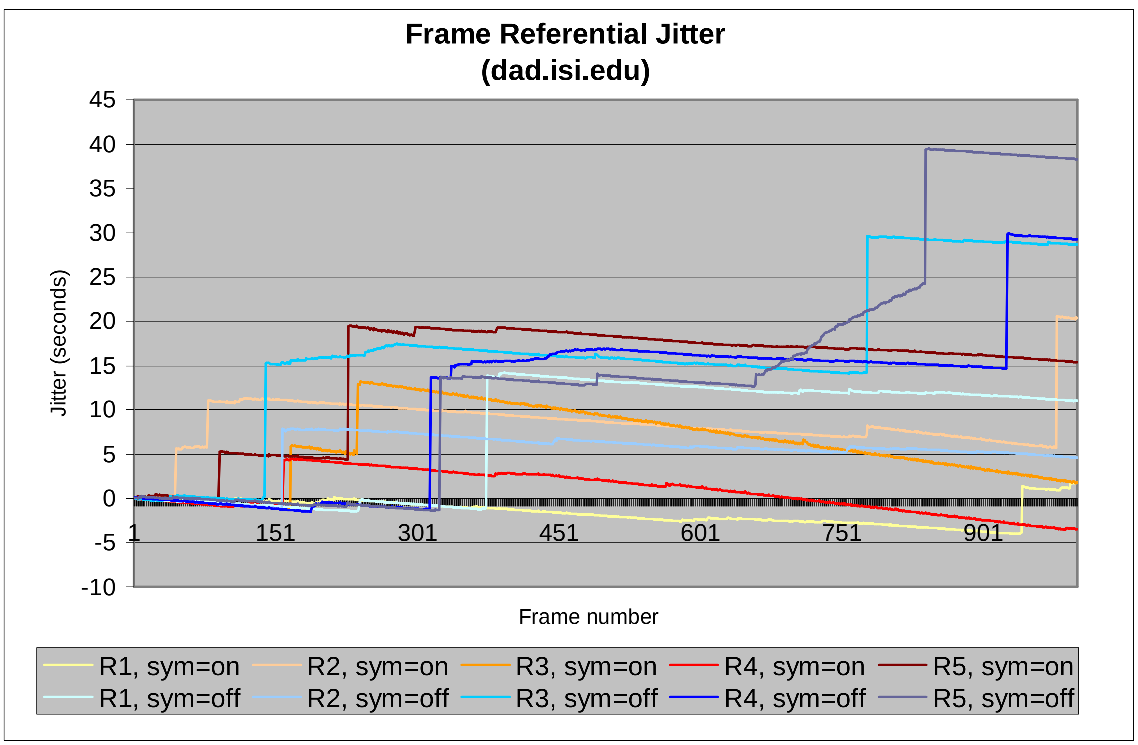

Figure-4 (D)

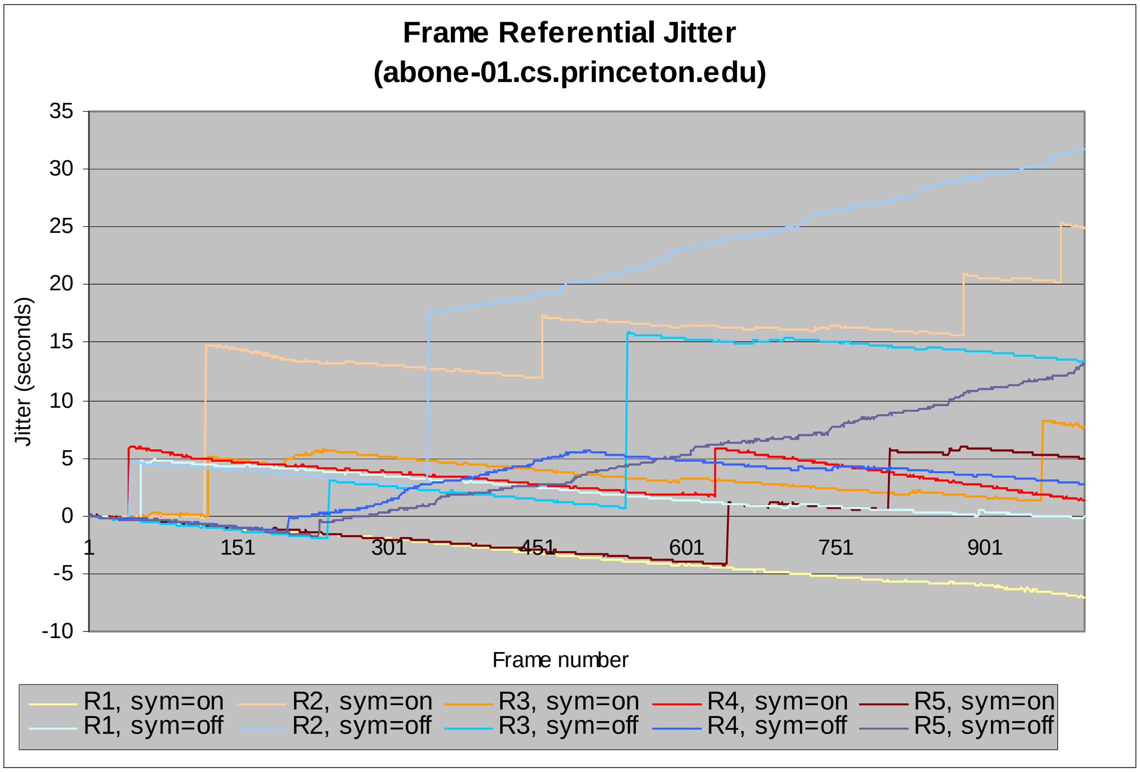

Figure-4 (E)

Figure-4. Per frame referential jitter. Negative jitter means that the frame arrived earlier than its ideal time. The classical TCP fell behind with each loss event.

References

[ABCS00] D. Andersen, D. Bansal, D. Curtis, S. Seshan, and H. Balakrishnan, System Support for Bandwidth Management and Content Adaptation in Internet Applications, Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, OSDI 2000, October 23-25, 2000 San Diego, California.

[AlPa99] Allman, Paxson, et al. TCP Congestion Control, RFC 2581, April 1999.

[BaRS99] Balakrishnan, H., Rahul, H., and Seshan, S., An Integrated Congestion Management Architecture for Internet Hosts," Proc. ACM SIGCOMM, Cambridge, MA, September 1999.pp.175-187.

[BrGM99] Hector M. Briceño, Steven Gortler and Leonard McMillan, NAIVE--network aware Internet video encoding, Proceedings of the seventh ACM international conference on Multimedia, October 30 - November 5, 1999, Orlando, FL USA, Pp 251-260.

[BrOP94] Brakmo, L.S., O'Malley, S.W., and Peterson, L.L, TCP Vegas: New Technique for Cogestion Detection and Avoidance, Proc. SIGCOMM' 94 Conf. ACM, pp-24-35, 1994.

[GuTe98] Wetherall, Guttag, Tennenhouse, "ANTS: A Tool kit for Building and Dynamically Deploying Network Protocols", IEEE OPENARCH'98, San Francisco, April 1998. Available at: http://www.tns.lcs.mit.edu/publications/openarch98.html

[ISO96] Information Technology- Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC International Standard 13818-2, June 1996.

[Jac88] Jacobson, V, "Congestion Avoidance and Control", Proc. SIGCOMM' 88, Conf, ACM, pp-214-329, 1988.

[Jac90] Jacobson, V., "Modified TCP Congestion Avoidance Algorithm," end2end-interest mailing list, April 30, 1990.

[KeWi00] Jun Ke and Carey Williamson / University of Saskatchewan, Towards a Rate-Based TCP Protocol for the Web, Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000.

[KhGu01] Javed I. Khan, Q. Gu, Network Aware Symbiotic Video Transcoding for Instream Rate Adaptation on Interactive Transport Control, IEEE International Symposium on Network Computing and Applications, IEEE NCA' 2001, October 8-10, 2001, Cambridge, MA, pp.201-213

[KHFH96] Keesman, Gertjan, Hellinghuizen, Robert Hoeksema, Fokke Heideman, Geert, Transcoding of MPEG bitstreams Signal Processing: Image Communication, Volume: 8, Issue: 6, pp. 481-500, September 1996,

[KYGP01] Javed I. Khan, Seung Su Yang, Qiong Gu, Darsan Patel, Patrick Mail, Oleg Komogortsev, Wansik Oh, and Zhong Guo Resource Adaptive Netcentric Systems: A case Study with SONET- a Self-Organizing Network Embedded Transcoder, Proceedings of the ACM Multimedia 2001, October 2001, Ottawa, Canada, pp617-620

[KhY01] Javed I. Khan & S. S. Yang, Resource Adaptive Nomadic Transcoding on Active Network, International Conference of Applied Informatics, AI 2001, February 19-22, 2001, Insbruck, Austria, [available at URL http://medianet.kent.edu/, also mirrored at http://bristi.facnet.mcs.kent.edu/medianet] (accepted).

[KhGR02] Javed I. Khan, Qiong Gu and Raid Zaghal, Symbiotic Video Streaming by Transport Feedback based quality rate selection, Proceedings of the 12th IEEE International Packet Video Workshop 2002, Pittsburg, PA, April 2002, http://www.pv2002.org .

[KhZ03a] Javed Khan and Raid Zaghal, "Event Model and Application Programming Interface of TCP-Interactive", Technical Report TR2003-02-03, Kent State University, [available at URL http://medianet.kent. edu/ technicalreports.html] March 2003.

[KhZ03c] Javed Khan and Raid Zaghal, "Performance Analysis of TCP Interactive based World Wide Video Streaming over ABone: Symbiotic Control and Video Quality", Technical Report TR2003-02-04, Kent State University, [available at URL http://medianet.kent. edu/ technicalreports.html] March 2003.

[KPOY01] Javed I. Khan, Darsan Patel, Wansik Oh, Seung-su Yang, Oleg Komogortsev, and Qiong Gu, Architectural Overview of Motion Vector Reuse Mechanism in MPEG-2 Transcoding, Technical Report TR2001-01-01, Kent State University, [available at URL http://medianet.kent. edu/ technicalreports.html, also mirrored at http:// bristi.facnet.mcs.kent.edu/medianet] January, 2001.

[Mpeg00] MPEG-2 Test Model 5, [URL: http://www.mpeg.org/MPEG/MSSG/tm5/, last retrieved December 25, 2000]

[PeDa00] 19 L. L. Peterson and B. S. Davie. Computer Networks, 2nd edition, Morgan-Kaufmann, 2000.

[PrCN00] Prashant Pradhan, Tzi-cker Chiueh and Anindya Neogi Aggregate TCP Congestion Control Using Multiple Network Probing, Proceedings of the The 20th International Conference on Distributed Computing Systems, ICDCS 2000. 2000.

[ReHE00] Reza Rejaie,Mark Handley, Deborah Estrin, Architectural Considerations for Playback of Quality Adaptive Video over the Internet, Proceedings of the IEEE International Conference on Networks (ICON'00), 2000.

[SiWo98] Dorgham Sisalem, Adam Wolisz, Towards TCP-Friendly Adaptive Multimedia Applications Based on RTP, Proceedings of the The Fourth IEEE Symposium on Computers and Communications, 1998.

[Tene96] Tanenbaum, A. Computer Networks, 3rd edition, Prentice Hall, 1996.

[TSSW97] Tennenhouse, D. L., J. Smith, D. Sincoskie, D. Wetherall & G. Minden., "A Survey of Active Network Research", IEEE Communications Magazine, Vol. 35, No. 1, Jan 97, pp 80-86

[Wolf97] Bernd E. Wolfinger, On the potential of FEC algorithms in building fault-tolerant distributed applications to support high QoS video communications, Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing , 1997, Pages 129 – 138