PERFORMANCE STUDIO OF MULTICAST VIDEO STREAMING USING SRMSH

Oscar Martínez Bonastre, Carlos Palau Salvador***

* Operations Research Centre. Miguel Hernandez University (Spain).
oscar.martinez@umh.es
** Communications Department. Polytechnic University of Valencia (Spain)
cpalau@dcom.upv.es

ABSTRACT

With the increasing deployment of multimedia real-time Internet applications, evaluating transport protocol metrics of Quality of Service (QoS) has gained rapidly increasing importance. In this paper, a novelty protocol named Scalable Reliable Multicast Stair Hybrid (SRMSH) is presented as new hybrid multiple layer mechanism for multicast congestion control providing detection and recovery loss. Then SRMSH is simulated with video streaming traffic source to measure fundamental components to real-time multimedia applications: Throughput, latency and jitter. Work is focused on performance analysis and results obtained from NS-2 traces clearly conclude that SRMSH exhibits interesting insights using these metrics of real-time multimedia applications.

1. INTRODUCTION

Multimedia systems [1] like multicast video applications or video-on-demand are particular real-time systems in which normally soft deadlines are to be fulfilled in order to provide a user-defined quality of service (QoS). Often they use multicast transport protocols so they cover different requirements in terms of congestion control. This paper proposes an innovative real-time protocol offering an hybrid congestion control algorithm called Scalable Reliable Multicast Stair Hybrid (SRMSH) which combines the benefits of simulating TCP's Additive Increase/multiplicative decrease with Rate-based (STAIR) [2] for layered multicast with Scalable Reliable Multicast (SRM) [3] in order to offer loss recovery methods if receivers detect loss events. So this paper introduces the SRMSH approach on a multimedia framework and examines fundamental metrics of QoS from any multimedia application like throughput, latency and jitter. Network Simulator-2 (NS-2) [4] is used for the experiments. The remainder of this paper is organized as follows.

In §II SRMSH mechanism is presented with their underlying multicast protocols. In §III multimedia simulation framework is described under SRMSH performance has been evaluated. Finally, §IV presents main conclusions and work in progress.

2. SCALABLE RELIABLE MULTICAST STAIR HYBRID

SRMSH approach has been specified using Communicating Real-Time State Machines (CRSMs) [5] as a formal method; result is a new hybrid congestion control mechanism which enables receivers following matters on real-time: (*i*) adapting their reception rate and (*ii*) offering loss recovery methods if they detect loss events. Next, SRMSH underlying real-time multicast protocols are presented, each one with different approach with congestion control. Then, we introduce the real-time SRMSH protocol.

2.1. Scalable Reliable Multicast protocol

SRM [3] is one of the most recognized and well known reliable multicast protocols which uses a real-time receiver oriented recovery mechanism. Although SRM uses reliable schemes, it does not consider flow control or congestion control mechanisms [6] i.e. SRM senders send at a fixed rate during all transmission period.

The protocol makes extensive use of IP multicast. The sender and receivers join an IP multicast group, and new messages are transmitted using IP multicast with its unreliable features. A receiver that detects data loss uses IP multicast to solicit a retransmission, and a participant receiving a solicitation uses IP multicast to repair the loss. The major innovation of SRM involves its use of stochastic mechanisms to avoid storms of REQUEST and REPAIRS when loss occurs. For example, a randomized delay is introduced before sending a request or repair, and the size of the delay is increased as a function of the estimated distance of the receiver from the sender. If a process *p* is waiting to solicit a

0-7803-9332-5/05/$20.00 ©2005 IEEE
Authorized licensed use limited to: Univerdad Miguel Hernandez. Downloaded on February 16,2024 at 11:38:06 UTC from IEEE Xplore. Restrictions apply.

retransmission for lost data, it will inhibit its own request in the event that a request from process *q* is received first. Similarly, a repair sent by one process will inhibit the sending of a repair by some other process. To ensure that lost data will be detected, all members of an SRM group send "*session*" messages periodically, at a frequency calibrated to keep the background overhead low and try to consume less bandwidth for this proposal.

Therefore, it's quite important to set appropriate timer parameter values for the SRM algorithm depending on the different scenarios because the final results are a function of them, more details in [3]. So the solution depends on the way that these values will change as network conditions changes. This motivates the development on the adaptive loss recovery algorithm, where the timer parameters are adjusted in response to past performance.

2.2. STAIR congestion control algorithm

STAIR [2] is a well refined and efficient real-time approach which combines the benefits of cumulative and non-cumulative layering. This mechanism, layered oriented, introduces a Stair Layer so named because the rates on these layers change dynamically over time, and in so doing resemble a staircase. This third layer, being positioned just above previous cumulative and non cumulative layers, is used to automatically emulate the additive/increase portion of AIMD congestion control, without the need of IGMP control traffic in order to reduce control traffic for congestion control. Then, one main difference in this approach from other congestion control algorithms deals with these dynamic Stair Layers in order to probe available bandwidth.

Different Stair Layers are used to accommodate additive increase for receivers with heterogeneous Round Trip Time (RTT) from the source. Thus, every Stair Layer own two main parameters: (*i*) RTT of t ms that is designated to emulate and (*ii*) Maximum rate R, measured in packets per t ms.

The rate transmitted on each Stair Layer is a cyclic step function with a minimum bandwidth of 1 packet per t ms, a maximum of R, a step size of 1 packet, and a stepping rate of 1 per RTT emulated. Upon reaching the maximum attainable rate, the Stair Layer recycles to a rate of 1 packet per RTT. A stair period of a given stair is defined as the duration of time that it takes the layer to iterate through one full cycle of rates.

In order to conduct AIMD congestion control, each receiver measures packet loss over stair period and if there is no loss detected, then the receiver performs an increase in its reception rate. Conversely, if there is packet loss event in a stair period, no method for recovery loss is offered and then one round of multiplicative decrease is performed, more details in [2].

As a result, in order to increase its subscription rates, it's vitally important that each receiver estimates or measures its RTT to subscribe to an appropriate Stair Layer so they must be configured carefully.

2.3. SRMSH protocol

SRMSH is focused on introducing SRM as loss recovery method into STAIR algorithm. Then, main motivation to implement this approach is to take advantage of the strong parts of each protocol and improve on the weak ones. This new hybrid congestion control schema has been specified using Communicating Real-Time State Machines (CRSMs) [5] as a formal method. For that purpose, SRMSH reaches synchronization in real-time applications to synchronize different data streams like video.

Firstly, each synchronization process (i.e. a SRMSH entity) is defined by a machine set pictured in figure 1. In addition, each machine manages a set of local variables and communication between machines is performed through global variables. *Inbuf* and *outbuf* deals with buffers between streaming source and user interface counting data context to transmit. Furthermore, each machine monitors SRMSH behaviour using assertions over timed traces of input-output events.

In view of that framework, SRMSH protocol is specified by a pair *{M, V}* where *V* includes a set of variables and *M* is a set of 6 machines with the following assigned tasks: (*i*) I_t, deals with host interface being responsible of start and finishing data transmission, (*ii*) I_r, deals with receiver interface for receiving data through buffers, (*iii*) T_d, will transmit data by different data flows as divided in different multicast addresses, (*iv*) R_d, will receive all transmitted or retransmitted data packets. Besides, this machine decides how many levels or multicast addresses to subscribe in order to receive information with a rate which is not going to produce congestion, (*v*) T_c, will transmit all control packets managing detection and recovery loss and (*vi*) R_c, will accept all control packets dealing with detection and recovery loss. In SRMSH, all members that belong to the same multicast session are able to act simultaneously as senders or receivers, always with the aim of support mechanism for loss recovery. Therefore, associated machine is the same for T_c and R_c. By last, several multicast addresses groups are used related to each subscription level. In particular, SRMSH approach has selected the multicast address associated with base cumulative layer defined in the STAIR scheme to send all the information related to detection and loss recovery. All these multicast addresses are depicted in figure 1 as outcoming or incoming channels *chan 1..n* respectively where *chan 1* is used for sending all the information related to detection and loss recovery.

Authorized licensed use limited to: Univerdad Miguel Hernandez. Downloaded on February 16,2024 at 11:38:06 UTC from IEEE Xplore. Restrictions apply.

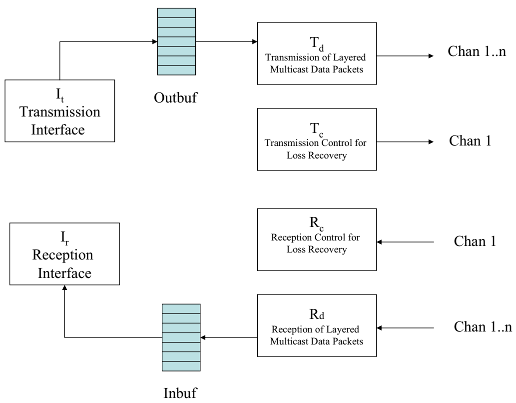

Figure 1. Machines Set for SRMSH.

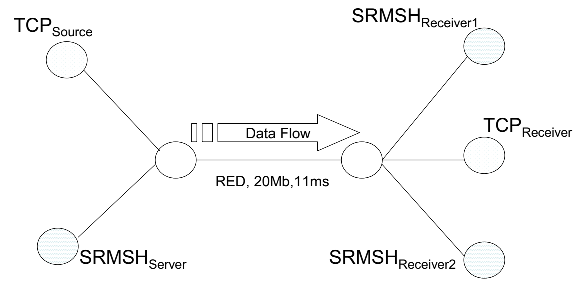

Figure 2. Network topology.

Thus, the emphasis is on requirements and design specifications methods for describing, predicting, and verifying the timing behaviour of SRMSH approach. By last, one of the most difficult goals for this approach dealt with global time of the network in order to synchronize with each machine timers for accuracy real-time requirements.

3. PERFORMANCE EVALUATION

SRMSH approach has been modelled and simulated using NS-2 [4] which has proved to be an excellent tool for simulating network behaviour.

Experiments were performed under following simple network topology showed in figure 2. A bottleneck configured with RED was introduced close to traffic sources and all other links, which not represent a bottleneck, have been configured Droptail as a router management policy, 10ms delay and 100Mb bandwidth. In order to simulate realistic networks where TCP traffic is always present, one TCP flow with TCP/Reno flavour has been added to node labelled TCP_{Source}. In node labelled $TCP_{Receiver}$, one sink was configured to receive such TCP traffic.

Traffic source like FTP has been added to node TCP_{Source}. Multicast video delivery has become important factor of today's Internet traffic so it was simulated on this framework by attaching to node labelled $SRMSH_{Server}$ a trace file produced by an encoding video of the Star Wars motion picture [7].

In this source model, packets are 512 bytes long and this trace had an average rate of approximately 512 kbps. Video traffic is received through multicast transmission by 2 SRSMH receiver agents added to nodes labelled $SRSMSH_{receiver1}$ and $SRSMSH_{receiver2}$ respectively. Simulation time was 100sec and during this period, in order to simulate heterogonous network where loss events are regular, a NS-2 Loss model was configured (t=70sec) on adjacent link to $SRMSH_{receiver2}$ which packet drop rate is a uniform distribution with a rate 0.1.

Firstly, figure 3 shows results dealing with SRMSH throughput flows. In this performance study, throughput was considered as a reception rate at receiver end for this scenario.

It's clearly shown whenever link state is accurately maintained at each node; both SRMSH receivers provide similar throughput performance.

However, when loss regular events appear (t=70sec) then inaccuracies are present so throughput is compromised with $SRMSH_{receiver2}$.

Nonetheless, such receiver can recover in real-time its losses as it receives "fresh" repair packets from other SRMSH members, sender included.

Secondly, the delay was considered as the time elapsed between the emission of the first packet of a data block by the transmitting $SRMSH_{Server}$ and its reception by the receiving end-system, i.e., $SRMSH_{receiver}$.

Concretely, the delay metric showed no significant difference among SRMSH receivers so the average value is represented in figure 4.

It's clearly shown that mean latency maintains a steady state which offers stability in order to receive real-time video streaming traffic packets.

We assume this steady value because of own multicast loss recovery algorithm, i.e., the request and repair could each one come from a node close to point of failure.

Thirdly, jitter was studied as variation in delay which results are depicted in figure 5. As before, average value is represented respectively.

Furthermore, average jitter is almost zero so avoid the impact of continuous quality changes through real-time video streaming traffic.

Authorized licensed use limited to: Univerdad Miguel Hernandez. Downloaded on February 16,2024 at 11:38:06 UTC from IEEE Xplore. Restrictions apply.

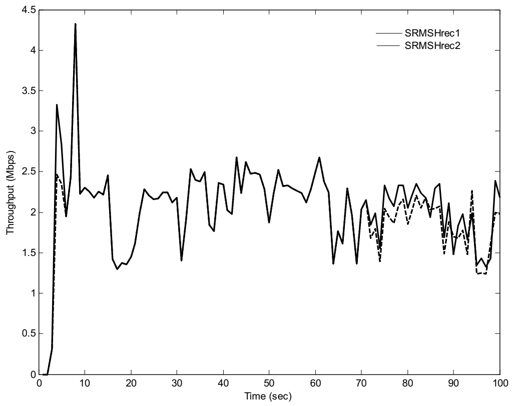

Figure 3. Throughput.

To summarize, other experiments were made using different simulation time and results showed similar oscillations.

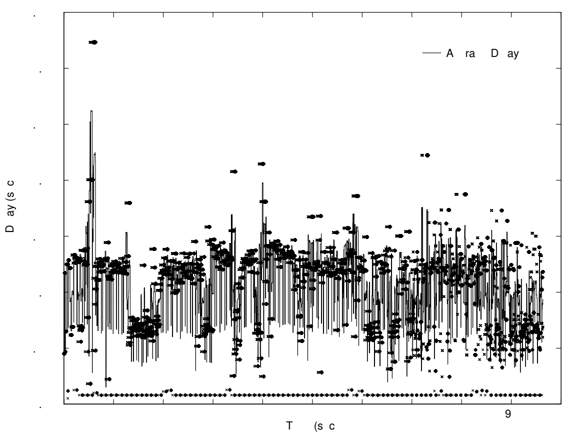

Figure 4. Delay.

4. CONCLUSIONS

As a result, following main conclusions are presented next: (*i*) experiments using NS-2 have demonstrated SRMSH flows generate a steady throughput, low latency and almost zero jitter so (*ii*) SRMSH is compliant with multimedia core QoS parameters. Finally, ongoing work in progress deals with getting other analysis with different parameters (scalability, recovery latency, overhead, implosion) and other multimedia metrics.

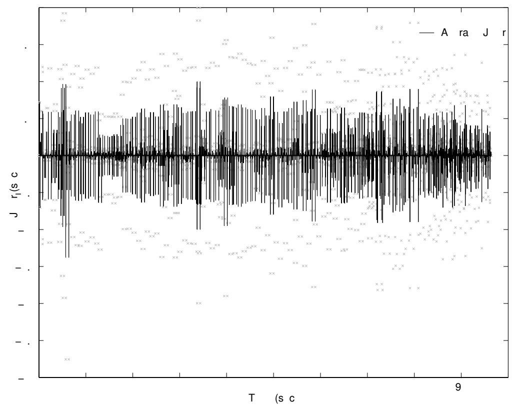

Figure 5. Jitter.

REFERENCES

[1] A. Matrawy, I. Lambadaris., "A Survey of Congestion Control Schemes for Multicast Video Applications", IEEE Communications Surveys & Tutorials, Volume 5, Issue 2, Fourth quarter, 2003.

[2] J. Byers, G. Kwon, "STAIR: Practical AIMD Multirate Multicast Congestion Control" in proc. of NGC, 2001.

[3] S. Floyd, V. Jacobson, et al., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", in IEEE/ACM Transactions on Networking. Volume 5, Number 6, pp. 784-803, December 1997.

[4] UCB/LBNL/VINT. Network Simulator ns-2. Available at http://www-mash.cs.berkeley.edu/ns/

[5] O. Martinez Bonastre, C. Palau Salvador, "A Synchronization Service Framework for multicast congestion control SRMSH approach using Communicating Real-Time State Machines", Technical Report, Operations Research Centre, October 2004.

[6] C. Hanle, M. Hoffman, "Performance comparison of Reliable Multicast Protocols using the network simulator NS-2", in IEEE LCN, 1998.

[7] F. Fitzek, M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation", IEEE Network, Vol. 15, Issue 6, pp 40-54, December 2001.

Authorized licensed use limited to: Univerdad Miguel Hernandez. Downloaded on February 16,2024 at 11:38:06 UTC from IEEE Xplore. Restrictions apply.