International Journal of Bifurcation and Chaos, Vol. 20, No. 9 (2010) 2965–2973

DOI: 10.1142/S0218127410027519

MULTICAST CONGESTION CONTROL SRMSH APPROACH USING COMMUNICATING REAL-TIME STATE MACHINES

O. M. BONASTRE* and S. NEVILLE†
Operations Research Center, Miguel Hernandez University,
Avda. de la Universidad, Elx 03202, Spain
**oscar.martinez@umh.es*
†stephen.neville@alu.umh.es

C. E. PALAU
Communications Department, Polytechnic University of Valencia,
C/Camino de Vera s/n, Valencia 46022, Spain
cpalau@dcom.upv.es

Received March 15, 2009; Revised October 25, 2009

Dedicated to the memory of Valery S. Melnik

New real-time applications frequently involve timing constraints related to accurate services from communication protocols. Concretely, real-time communication protocols utilize timers to implement these constraints between system event occurrences. In this context, the study of congestion control for Internet reliable multicast is at present an active research field related to real-time protocols. In this paper, the authors present an innovative real-time transport protocol named *Scalable Reliable Multicast Stair Hybrid* (SRMSH) as new hybrid multiple layer mechanism for multicast congestion control providing detection and recovery loss. This work is focused on formal specification of SRMSH approach using Communicating Real-Time State Machines as a formal method. Besides, SRMSH validation is presented within a formal proof framework in order to check the functional safety and liveness properties. As a result, authors outline a dynamical system framework in order to model behavior of their presented solution.

Keywords: Formal methods; real-time transport protocols; multicast congestion control; reliable multicast; layered multicast; dynamical systems.

1. Introduction

Real-time protocols have a strong multidisciplinary nature that integrate information and communication technology. In this context, many methodologies for modeling and designing real-time protocols include formal methods [Mathai, 2001; Babich & Deotto, 2002; Laplante, 2004] for specification and analysis dealing with describing, predicting, and verifying their timing behavior. As a result, modeling and design of complex real-time protocols require accurate notations for describing concurrency, communication between independent machines and timing properties. In this framework, new real-time applications frequently involve timing constraints related to accurate services from transport protocols.

Concretely, the study of congestion control for Internet reliable multicast is at present an active research field for real-time systems.

In this paper, an innovative real-time protocol is presented like an hybrid congestion control algorithm named *Scalable Reliable Multicast Stair Hybrid* (SRMSH) which combines the benefits of simulating TCP's Additive Increase/Multiplicative Decrease with Rate-based (STAIR) [Byers & Kwon, 2001] for layered multicast with Scalable Reliable Multicast (SRM) [Floyd *et al.*, 1997] in order to offer loss recovery methods if receivers detect loss events.

First and foremost, SRMSH approach is formally specified using standard state machine notations from Communicating Real-Time State Machines (CRSMs) in order to convey its own real-time system characteristics and constraints.

The remainder of this work is organized as follows. Section 2 presents the state-of-the-art related with multicast congestion control.

Then, Sec. 3 introduces underlying multicast protocols of SRMSH real-time mechanism, each one with different approach with congestion control. Section 4 presents SRMSH specification using CRSMs as a formal method. Thus, authors discuss the corresponding transition labels for each SRMSH state machine with their associated predicates, actions and deadlines. Section 5 validates the SRMSH formal specification. Section 6 outlines a dynamical system framework in order to model the behavior of the presented solution. Finally, Sec. 7 presents the main conclusions and work in progress.

2. Related Work with Multicast Congestion Control

In the past few years, numerous research studies [Obrazka, 1988; Golestani & Sabnani, 1988; Widmer *et al.*, 2001] have been carried out to explore how to support multicast in heterogeneous networking environments.

Especially, congestion control is a major issue in the design of real-time transport protocols which run on top of the Internet multicast service. As a consequence, most of the current studies and proposals for congestion control of multicast traffic in the Internet deal with two problems in common: how they use a rate-based mechanism for traffic regulation and how they try to detect and recover from loss events.

Rate-based mechanisms are divided into two different categories: single rate and multiple rate. The main difference between them is to allow modifying rate transmission during multicast session period (multiple rate) or not (single rate) for different receiver groups.

On the other hand, for multicast session behavior, two different choices are offered: sender-driven or receiver-driven. Both of them deal with congestion control decisions which are resolved by the source (senderdriven) [Kasera *et al.*, 2000; Chiu *et al.*, 2002; Widmer & Handley, 2003; Rhee *et al.*, 1999, 2000; Rizzo, 2000; Sisalem & Wolisz, 2000; Speakman *et al.*, 2001; Yano & McCnanne, 2000; Mongomery, 1997] or by each receiver separately (receiver-driven) [Byers *et al.*, 2000; Vicisano *et al.*, 1998; Floyd *et al.*, 1997; Byers & Kwon, 2004; Holbrook *et al.*, 1995].

In single rate approaches [Widmer & Handley, 2003; Rizzo, 2000], all receivers have the same reception rate. Frequently, this rate value is according to the slowest receiver rate. With this proposal it is not necessary to send information through different flow data and coding. Nonetheless, limitation factors occur when receivers are in huge networks inside diverse heterogeneous groups.

The typical drawback is when the reception rate is extremely low due to the receiver being too slow, and thus affecting all others receivers.

Multiple rate approaches [Byers *et al.*, 2000; Byers *et al.*, 2001; Vicisano *et al.*, 1998; Floyd *et al.*, 1997; Byers & Kwon, 2004], use multiple multicast groups in order to transmit content at different rates, generally classified in cumulative [Vicisano *et al.*, 1998] or not cumulative layers [Byers *et al.*, 2001].

In more detail, these algorithms which are used for audio/video transmission [Matrawy & Lambadaris, 2004] through the Internet employ layered multicast.

On the contrary, single rate proposal uses this choice to handle numerous and heterogeneous groups of receivers. So far, proposals based on layered multicast use static or dynamic layers. In static layer approaches [Vicisano *et al.*, 1998], the sending rate for each layer maintains the same value during all transmission period.

One disadvantage of this method is seen when sometimes the receiver does not own appropriate information in order to join more layers, so the reception rate could change in an abrupt manner. Different algorithms [Byers *et al.*, 2000] use dynamic layers which allow the receivers to be coordinated in a better way whenever they are behind the same bottleneck or even reduce the number of Internet Group Management Protocol (IGMP) messages.

Obviously, these actions are taken in order to improve the reception rate that avoids congestion.

Last but not least, another significant issue with congestion control is loss recovery and detection. Well-known proposals [Birman *et al.*, 1999; Floyd *et al.*, 1997; Holbrook *et al.*, 1995; Lin & Paul, 1996; Byers & Kwon, 2001] offer receivers smart solutions when losses occur during multicast transmission. So the key factor is how to detect losses and which actions will be taken by the receiver for each loss event.

Within this multicast framework, as introduced before, this paper proposes an innovative real-time protocol by offering an hybrid congestion control algorithm named *Scalable Reliable Multicast Stair Hybrid* (SRMSH).

Our approach combines the benefits of simulating TCP's Additive Increase/Multiplicative Decrease with Rate-based (STAIR) [Byers & Kwon, 2001] for layered multicast with Scalable Reliable Multicast (SRM) [Floyd *et al.*, 1997] in order to offer loss recovery methods if receivers detect loss events.

3. Underlying Protocols for SRMSH Approach

In this section, SRMSH underlying real-time multicast protocols are presented, each one with a different approach to congestion control. Later, SRMSH formal specification is introduced in Sec. 4.

3.1. *Scalable reliable multicast*

Scalable Reliable Multicast (SRM) [Floyd *et al.*, 1997] is one of the most recognized and well-known reliable multicast protocols which uses a receiver oriented recovery mechanism on real-time. While SRM uses reliable schemes, it does not consider flow control or congestion control mechanisms [Hanle *et al.*, 1998], i.e. SRM senders transmit at a fixed rate during the entire 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 request a retransmission, and a participant receiving a solicitation uses IP multicast to repair the loss.

To ensure that lost data will be detected, all members from the SRM group send session (*Heartbeat*) messages periodically, that reports the sequence number state for active sources.

Members can also use session messages in SRM to determine the current participants of the session and then for dynamically adjusting the generation rate of session messages in proportion to the multicast group size.

During the multicast session, SRM members who detect a loss wait for a random time and then multicast their repair request messages, to suppress requests from other members sharing that loss. When a host A detects a loss, it schedules a repair request for a random time in the future.

When the request timer expires, host A multicasts a request message for the missing data, and doubles the request timer to wait for the repair. The interval over which the request timer is set is a function of the member's estimated distance to the source of the packet. Thus, if host A receives a request message for the missing data before its own request timer for that data expires, then host A does a (random) exponential backoff, and resets its request timer.

When some other host B (where B may be the original source S) receives a request message from A that host B is capable of answering, host B sets a repair timer.

If host B receives a repair message for the missing data before its repair timer expires, then host B cancels its repair timer. Otherwise, when host B's repair timer expires, host B multicasts the repair.

Due to the probabilistic nature of these algorithms, it is not unusual for a dropped packet to be followed by more than one request.

When two or more hosts generate a request for the same data at roughly the same time, the network includes redundant control traffic (i.e. wasted bandwidth) and the colliding participants should increase the spread in their retransmission distribution to avoid similar collisions in the future.

Because there can be more than one request, a host could receive a duplicate request immediately after sending a repair, or immediately after receiving a repair in response to its own earlier request.

In order to prevent duplicate requests from triggering a responding set of duplicate repairs, host B ignores requests for data D for a period of time after sending or receiving a repair for that data D, where host S is either the original source of data D or the source of the first request.

The major innovation of SRM involves its use of stochastic mechanisms to avoid storms of request and repairs when loss occurs.

Consequently, its quite important to set the appropriate timer parameter values for the SRM algorithm depending on the different scenarios because the final results are a function of them.

So the solution depends on the way that these values will change as the network conditions change. In conclusion, this motivates the development of the adaptive loss recovery algorithm, where the timer parameters are adjusted in response to past performance.

More details about parameters and timers of the request, repair algorithms are found in [Floyd *et al.*, 1997]. In the next subsection, the authors will introduce a well-known approach which offers multicast based solution to the congestion control problem.

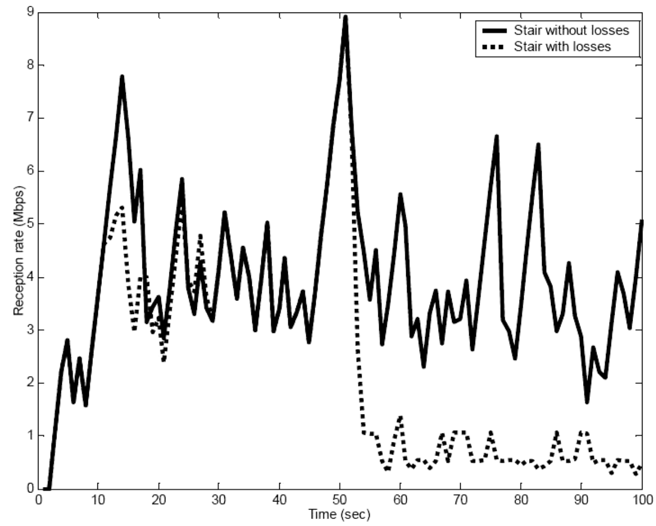

Fig. 1. Stair drop-to-zero problem.

3.2. *Stair congestion control algorithm*

STAIR [Byers & Kwon, 2001] is a well refined and efficient real-time approach which combines the benefits of cumulative and noncumulative layering. This mechanism, layered and oriented, introduces a Stair Layer so named because the rates on these layers change dynamically over time, and in so doing resemble a staircase. Dynamic layers have been used by [Byers *et al.*, 2000] to probe the available bandwidth so one important difference in this approach from other congestion control algorithms deals with these dynamic Stair Layers.

This third layer, being positioned just above the previous cumulative and noncumulative layers, is used to automatically emulate the additive/increase portion of Additive Increase Multiplicative Decrease (AIMD) congestion control, without the need of IGMP control traffic in order to reduce the control traffic for congestion control. Different Stair Layers are used to accommodate additive increase for receivers with heterogeneous Round Trip Time (RTT) from the source. Thus, every Stair Layer owns two main parameters:

- RTT of t ms that is designated to emulate.
- Maximum rate R.

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 the packet loss over the stair period and if there is no loss detected, then the receiver performs an increase in its reception rate. Conversely, if there is a packet loss event in a stair period, no method for recovery loss is offered and then one round of multiplicative decrease is performed. Therefore, STAIR has a lack of losses recovery mechanism, the main STAIR weakness, the drop-to-zero problem (see Fig. 1), occurs when a receiver detects regular loss events and immediately the STAIR algorithm performs multiplicative decrease dropping to a low reception rate.

Additionally, in order to increase its subscription rates, it is vitally important that each receiver estimates or measures its RTT to subscribe to an appropriate Stair Layer so they must be configured carefully. More details in [Byers & Kwon, 2001].

4. Scalable Reliable Multicast Stair Hybrid Formal Specification

The SRMSH approach has been specified using Communicating Real-Time State Machines as a formal method [Babich & Deotto, 2002] and simulated with NS2 [Babich & Deotto, 2002] successfully. The result is a new hybrid congestion control mechanism which enables the receivers to follow matters on real-time:

- Adapting their reception rate.
- Offering loss recovery methods if they detect loss events.

Therefore, the SRMSH approach is focused on introducing SRM as loss recovery method into the STAIR algorithm. Consequently, the main goal of implementing this approach is to maximize the main advantages of both protocols and enforce the weakness of each one. For that purpose, SRMSH reaches synchronization in real-time applications to synchronize different data streams. With these premises, the research team has specified a practical synchronization protocol included in SRMSH formal specification. Firstly, each synchronization process (i.e. a SRMSH entity) is defined by a machine set pictured in Fig. 2. In this framework, each machine manages a set of local variables and communication between the machines is performed through global variables. In addition, *inbuf* and *outbuf* deals with buffers between streaming source and user interface counting the data context to transmit.

In view of that framework, the SRMSH protocol is specified by a pair M, V where V includes a set of global variables and M is a set of six machines with the following assigned tasks:

- I_t, deals with the host interface being responsible for start and finishing data transmission.
- I_r, deals with the receiver interface for receiving the data through buffers.
- T_d, will transmit data by different data flows as divided into different multicast addresses.
- R_d, will receive all transmitted or retransmitted data packets. Besides, this machine decides how many levels or multicast addresses are to be subscribed in order to receive information with a rate which is not going to produce congestion.
- T_c, will transmit all control packets managing detection and recovery loss.
- 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 supporting the mechanism for loss recovery. Then, the associated machine is the same for T_c and R_c. Lastly, several multicast addresses groups are used to each subscription level. In particular, the SRMSH approach selects 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 Fig. 2 as outcoming or incoming channels *chan 1* $\cdots$ n where *chan 1* is used for sending all the information related to detection and loss recovery.

Secondly, Fig. 3 depicts the state machines diagram for SRMSH protocol where each machine, as introduced in Fig. 2, monitors SRMSH behaviour using assertions over timed traces of input–output events.

This new hybrid congestion control schema has been specified using CRSMs as a formal method. Consequently, the emphasis is on requirements and design specifications methods for describing,

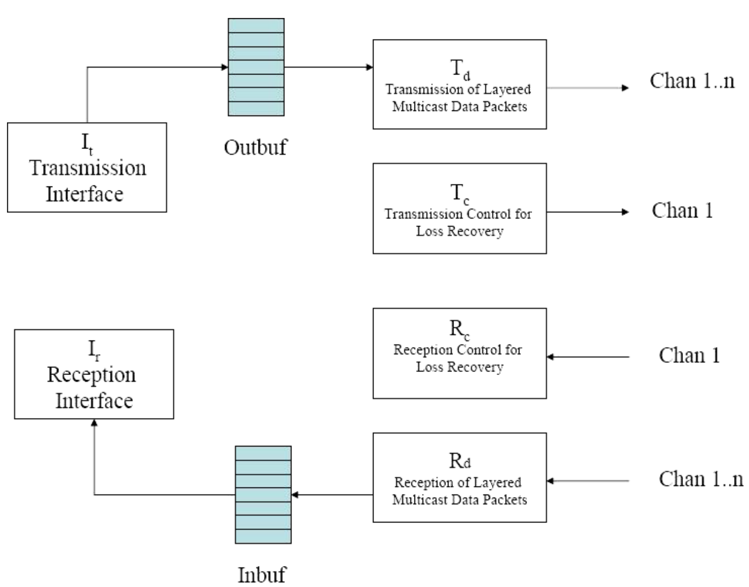

Fig. 2. Machines set for SRMSH.

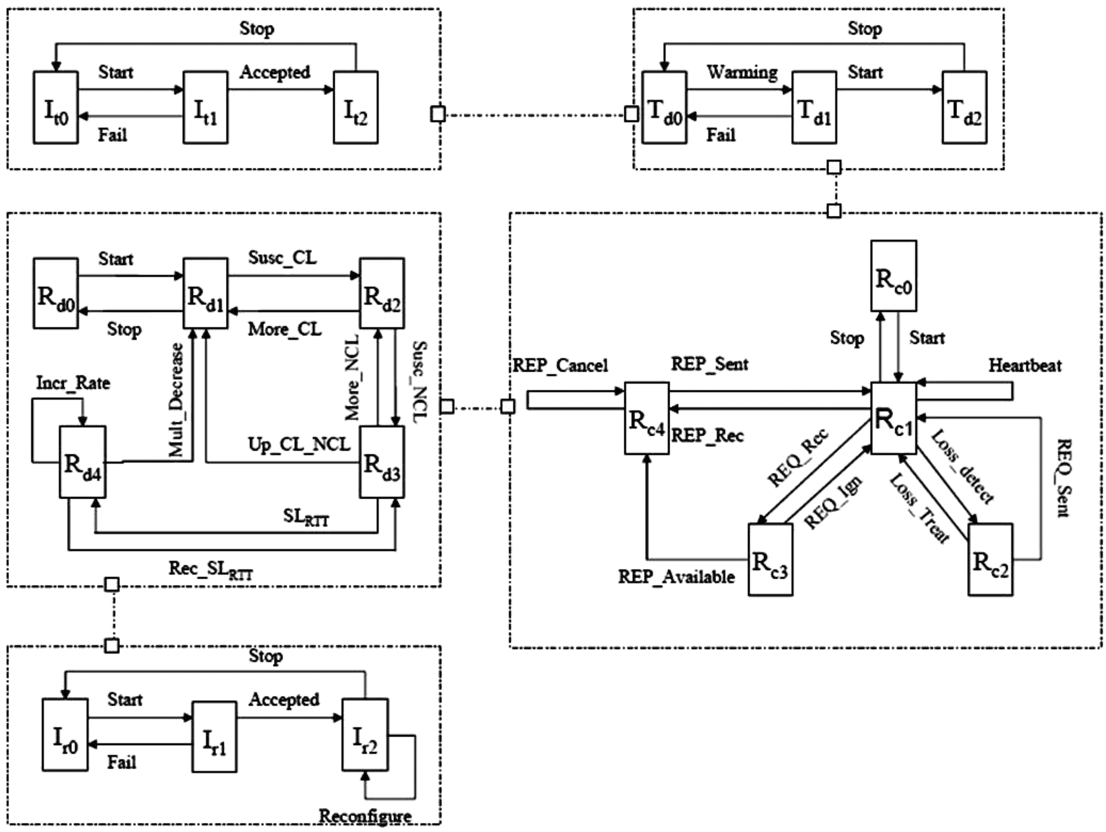

Fig. 3. State machines diagram for SRMSH protocol.

predicting, and verifying the timing behavior of SRMSH approach. Thus, Fig. 3 depicts each machine, I_t-T_d-R_c-R_d-I_r respectively. Concretely, it includes a transition label with its associated predicates, actions and time deadlines, respectively. Thus, each state machine is defined by (S_i, S_0, L_i, N_i, T_i) where:

- S_i, finite set of states.
- S_0, initial state included in S_i.
- L_i, local variables of the machine i.
- N_i, set of transition labels of the machine i. Associated with each transition exists a tupla (p, a, t) where p is the predicate or transition condition, a is the action associated with the transition and t is a time deadline to perform the transition.
- T_i, transition function defined as: $T_i : S_i \times N_i \rightarrow S_i$.

Lastly, the communication between state machines is done using global variables. Concretely, for specification purposes, a generic format message is defined with different message types for congestion control, detection and recovery loss. More details in [Martinez Bonastre, 2005].

5. Validation

Necessarily, every real-time system should pass successfully a validation of its own formal specification [Laplante, 2004; Mathai, 2001]. For that reason, validation determines whether the real-time system development achieves the requirements established during the previous phase of formal specification. Thus, model checking is a formal method that can be used to perform analysis of requirement specifications, even partial ones. The goal is also to look for errors, not only prove accuracy. As a result, this formal methodology based on state space uses state machines to test fundamental properties for any real-time protocol like safety and liveness [Udaya Shankar & Lam Simon, 1987], that is

- Safety property means that all states can be checked with reachability analysis and specifies the absence of certain undesirable events.
- Liveness property implies that the approximate time is as close as possible to real-time. That is to specify the progress of a computation to imply that something would happen in time units that again implies the liveness property termination.

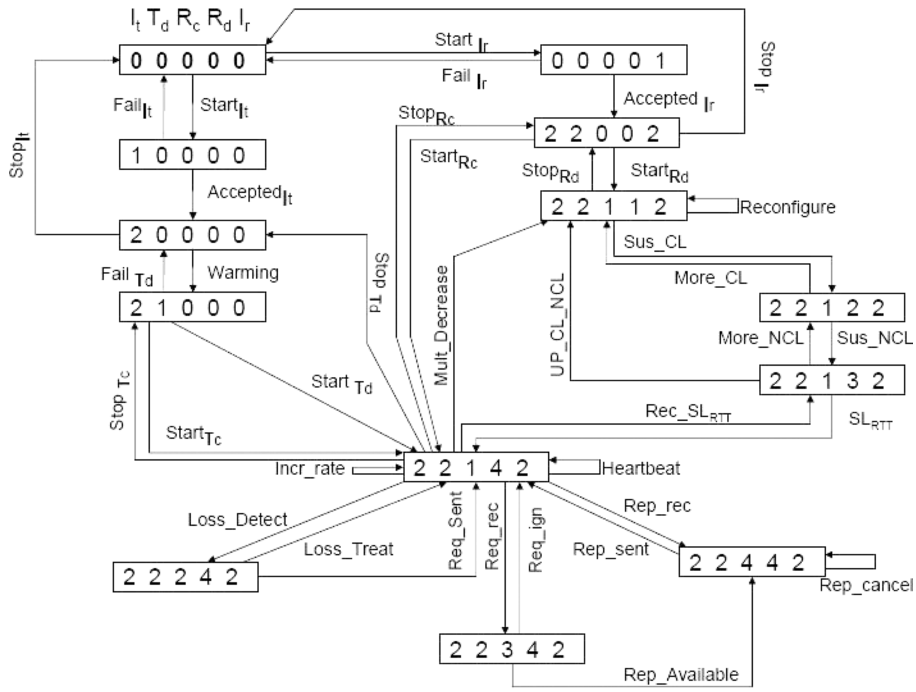

Fig. 4. Validation scheme.

To validate SRMSH formal specification, the first step involves building a state model of the SRMSH congestion control schema, in this case, using state space. The model is obtained by representing this state space for assessing the potential for automated validation. Besides, this method describes the formalism in which safety and liveness properties can both be described. About SRMSH validation analysis, Fig. 4 shows nodes representing global states of SRMSH machines set where links determine transitions communicating all SRMSH state machines. Transitions are renamed by the corresponding state machine when the label name is the same. Firstly, all desirable states of the transitions set represented in Fig. 4 are reached eventually, i.e. desirable states must be reached within known deadlines. Then, state progress is always achieved so that SRMSH approach accomplishes liveness. Secondly, time constraints between each transition can be specified and verified as safety assertions. In addition, there are sufficient conditions to avoid the possibility that arbitrary constraints on the enabling conditions of time events could cause them to deadlock. These conditions correspond to time constraints that are implementable within individual processes, being one of the most difficult goals for this approach, related to the global time of the network in order to synchronize with each machine timers.

6. Dynamical System Framework

A discrete dynamical system can be characterized as an iterated function [Steele, 2006]. Thus, a discrete dynamical system consists of a recurrence relation (or difference equation) describing some relationship, pure or applied, with a given set of initial conditions [Blanchard *et al.*, 2005]. Due to heuristic reasons achieved through simulations of our approach [Martinez Bonastre & Palau, 2009], we think that our developed real time protocol could be modeled by a discrete dynamical system, i.e. where time changes in discrete steps. Additionally, we have demonstrated in prior sections that SRMSH changes its behavior continuously depending on the discrete real-time values, i.e. SRMSH follows discrete behavior according to associated real-time rules. Consequently, this allows us to discuss key concepts of dynamical systems like equilibrium points, stability, limit cycles and other issues. That is, we achieve the fundamental factors that govern the qualitative behavior of SRMSH as a discrete dynamical system.

In this framework we propose to introduce methods for the analysis of discrete dynamical systems within the SRMSH. Firstly, the SRMSH model includes specific states which evolve after a period of time, i.e. periodic orbits we name Stair Layers as we have introduced in Sec. 4. Secondly, our model includes points which should be stable through the time, i.e. whenever SRMSH starts the reception rate will converge to the total sum of all subscribed layers (cumulative and noncumulative) as introduced in Sec. 4 also. Then, authors focused the analysis initially on the derivation of basic propositions on the factors that determine the local and global stability of discrete dynamical systems in the context of SRMSH.

7. Conclusions

This paper has presented SRMSH as an innovative real-time transport protocol characterized by a new hybrid multiple layer mechanism for multicast congestion control providing detection and recovery loss. Besides, this approach has been formally introduced as a novelty real-time multicast protocol which adapts by itself to the network conditions reaching congestion control requirements.

As a main conclusion, this real-time mechanism could be be modeled as a dynamical system, i.e. reaching a formal representation of its main features initially. This representation has demonstrated a behavioural, temporal logic which emulates the coarse-grain behavior of the SRMSH schema. About work in progress, SRMSH continues being prototyped according to the dynamical system rules.

Acknowledgments

This work has been supported by the Government of Spain through MTM2008-06778-C02-01 and European Commission through FP7-ICT-2007-2.

References

Babich, F. & Deotto, L. [2002] "Formal methods for specification and analysis of communication protocols," *IEEE Commun. Surv. Tutor.* **4**, 2–20.

Birman, K. *et al.* [1999] "Bimodal multicast," *ACM Trans. Comput. Syst.* **17**, 41–88.

Blanchard, P., Devaney, R. & Hall, F. [2005] *Differential Equations*, 3rd edition (Thomson Editorial).

Byers, J. *et al.* [2000] "Flid-dl: Congestion control for layered multicast," *Second Int Workshop on Networked Group Communication*, pp. 71–81.

Byers, J. & Kwon, G. [2001] "STAIR: Practical AIMD multirate multicast congestion control," *Third Int. Workshop on Networked Group Communication*, pp. 100–112.

Byers, J., Luby, M. & Mitzenmacher, M. [2001] "Fine-grained layered multicast," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 1143–1151.

Byers, J., Luby, M. & Mitzenmacher, M. [2002] "A digital fountain approach to asynchronous reliable multicast," *IEEE J-SAC, Special Issue on Network Support for Multicast Communication* **20**, 1528–1540.

Byers, J. & Kwon, G. [2004] "ROMA: Reliable overlay multicast with loosely coupled TCP connections," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 141–149.

Chaintreau, A., Baccelli, F. & Diot, C. [2002] "Impact of TCP-like congestion control on the throughput of multicast groups," *IEEE/ACM Trans. Networking* **10**, 500–512.

Chiu, D. *et al.* [2002] "A congestion control algorithm for tree-based reliable multicast protocols," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 340–358.

Floyd, S. *et al.* [1997] "SRM: Scalable reliable multicast. A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Trans. Networking* **5**, 784–803.

Golestani, S. J. & Sabnani, K. K. [1999] "Fundamental observations on multicast congestion control in the internet," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 990–1000.

Hanle, C. & Hoffman, M. [1998] "Performance comparison of reliable multicast protocols using the network simulator NS-2," *Proc. IEEE Local Computer Networks* (*LCN*), pp. 222–237.

Holbrook, H. W. *et al.* [1995] "Log-based receiver-reliable multicast for distributed interactive simulation," *Proc. ACM SIGCOMM*, pp. 328–341.

Kasera, S. *et al.* [2000] "Scalable fair reliable multicast using active services," *IEEE Netw. Mag.* (*Special Issue on Multicast*) **14**, 48–57.

Laplante, P. [2004] *Real-Time Systems Design and Analysis*, 3rd edition (IEEE Press–John Wiley).

Lin, J. C. & Paul, S. [1996] "Rmtp: A reliable multicast transport protocol," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 1414–1424.

Martinez Bonastre, O. [2005] "A synchronization service framework for multicast congestion control SRMSH approach using communicating real-time state machines," Technical Report, Operations Research Centre, on line available at http://cio.umh.es

Martinez Bonastre, O. & Palau, C. [2009] "A collaborative mobile architecture for multicast live-streaming social networks," *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 1764–1767.

Mathai, J. [2001] *Real-Time Systems Specification, Verification and Analysis,* Prentice Hall publications.

Matrawy, A. & Lambadaris, I. [2004] "A survey of congestion control schemes for multicast video applications," *IEEE Commun. Surv. Tutor.* **5**, 22–31.

Mongomery, T. [1997] "A loss tolerant rate controller for reliable multicast," Technical Report, *NASA-IVV*, 97–111.

Network Simulator ns-2 [2007] On line available at http://www-mash.cs.berkeley.edu/ns/

Obrazka, K. [1988] "Multicast transport protocols: A survey and taxonomy," *IEEE Commun. Mag.* **36**, 94–102.

Rhee, I. *et al.* [1999] "MTCP: Scalable tcp-like congestion control," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 1265–1273.

Rhee, I. *et al.* [2000] "Tear: TCP emulation at receivers — Flow control for multimedia steaming," Technical Report, Department of Computer Science, NCSU.

Rizzo, L. [2000] "PGMCC single rate multicast congestion control: Protocol specification," *Proc. ACM SIGCOM*, pp. 17–28.

Rubenstein, D., Kurose, J. & Towsley, D. [2002] "The impact of multicast layering on network fairness," *IEEE/ACM Trans. Networking* **39**, 169–182.

Sisalem, D. & Wolisz, A. [2000] "Mlda: A Tcp-friendly congestion control framework for heterogeneous multicast environments," *Eighth Int. Workshop on Quality of Service*, pp. 65–74.

Speakman, T. *et al.* [2001] "PGM reliable transport protocol," Internet Draft, draft-speakman-pgm-spec-06.txt, RFC 3208.

Steele, T. [2006] "Continuity and chaos in discrete dynamical systems," *Aequationes Mathematicae* **71**, 300–310.

Udaya Shankar, A. & Lam Simon, S. [1987] "Time-dependent distributed systems: Proving safety, liveness, and real-time properties," *Distrib. Comput. J.* **2**, 61–79.

Vicisano, L., Rizzo, L. & Crowcroft, J. [1998] "TCP-like congestion control for layered multicast data transfer," *Proc. IEEE Computer and Communication Societies* (*INFOCOM*), pp. 996–1003.

Widmer, J., Denda, R. & Mauve, M. [2001] "A survey of TCP-friendly congestion control," *IEEE Netw. Mag.* **15**, 28–37.

Widmer, J. & Handley, M. [2003] "Extending equation-based congestion control to multicast applications," *Proc. ACM Special Interest Group on Data Communication* (*SIGCOMM*), pp. 275–285.

Yano, K. & McCnanne, S. [2000] "A window-based congestion control for reliable multicast based on TCP dynamics," *Eighth ACM Int. Conf. Multimedia*, pp. 249–258.