An Approach to Optimize Multicast Transport Protocols

Dávid Tegze
Budapest University of Technology and Economics, Hungary

Mihály Orosz
Budapest University of Technology and Economics, Hungary

Gábor Hosszú
Budapest University of Technology and Economics, Hungary

Ferenc Kovács
Budapest University of Technology and Economics, Hungary

INTRODUCTION

The article presents an approach to optimize the multicast transport protocols. The main constraint of this procedure is the orthogonality (linear independence) of protocol parameters. Protocol parameters are variables defined for protocol classes, where the possible values of each parameter are protocol mechanisms, which serve the same goal in the multicast transport protocol. A multi-dimensional hyperspace of protocol parameters is stated, as a mathematical model of the optimization process where every transport protocol is represented as an individual point. A multicast transport *Sim*ulator for multi*Cast* (*SimCast*) has been developed to describe the performance of the transport protocols and to simulate the operation of these protocols for reliable multicasting. The simulator supports the protocol analysis in the hyperspace of protocol parameters.

BACKGROUND

Reliability is one of the most important features of all multimedia applications. This requirement may be especially critical in case of multicast, where the timely correction or resending of lost data is even more difficult because of the large volume of data to be transferred. Multimedia applications make multicast an active area of research. Multicasting is the one-to-many group communication way. For this purpose the IP-multicast transport is the preferred mechanism (Hosszú, 2005).

Since most of the multicast applications are media-related software, for example, media conference, voice distribution, shared whiteboard, and various collaborative media tools, they need more reliability than the best-effort delivery of Internet protocol (Adamson & Macker, 2001; Luby, & Goyal, 2004). In order to increase the reliability of multicast applications additional *multicast transport protocols* are used to achieve the required level of reliability (Whetten & Taskale, 2000). Such a protocol is the *NORM*: *N*ACK-*O*riented *R*eliable *M*ulticast Protocol (Adamson, Bormann, Handley & Macker, 2004a, 2004b).

It is hard to compare the various protocol mechanisms implemented in different protocols. Therefore, the modularly structured simulator *SimCast* (*Sim*ulator for multi*Cast*) is developed for traffic analysis of unicast (one-to-one) and multicast (one-to-many) streams (Orosz & Tegze, 2001). To carry out the necessary analysis of the unicast and multicast traffic, a well usable simulation program should be applied in order to present statistically correct results for multicast data transfer. The reason of developing a new, custom simulator instead of using a standard framework like *ns* (Breslau, Estrin, Fall, Floyd, Heidemann, Helmy, Huang, McCanne, Varadhan, Xu, and Yu, 2000) is that the architecture of the *SimCast* simulator is optimized for transport layer modeling and, due to its modular design, it is relatively easy to integrate new protocol mechanisms in it.

DECOMPOSITION OF THE MULTICAST TRANSPORT PROTOCOLS

Multicast transport protocols have many different properties for data delivery. These attributes can be represented by the previously mentioned *protocol parameters* (Hosszú, 2005). Each protocol parameter specifies different reliability mechanisms for the same delivery attribute. Such a protocol parameter is, for instance, the repair method, which can have the values like "retransmission", "forward-error correction", "interleaving", or different ways of "local receiver-based repairs" (Luby & Vicisano, 2004). Another parameter is the acknowledgement type, which could hold the possible values "tree-based", "ring-based" or a "simple direct form".

Various applications have different reliability requirements and, therefore, these protocol parameters should be optimized in order to determinate the best-suited multicast transport protocol for a given application. However, to use any mathematical optimization method for the selection of the protocol parameters, a linearly independent (in other words *orthogonal*) set of parameters should be applied. For this purpose, a hyperspace of protocol parameters is created where each individual transport protocol corresponds to a point of this multi-dimensional space. The optimization procedure finds the most appropriate point in this space to provide the best performance for multicast content delivery. The possible values of the protocol parameters (which are classes of various mechanisms) are the realizations of specific protocol functionalities. A *quasi-orthogonal* subset of the protocol parameters and their possible values are presented in *Table 1*. These parameters represent the well-known reliability mechanisms of transport protocols.

To carry out a correct optimization procedure on the appropriately selected protocol parameters, a well usable simulation program should be applied in order to obtain statistically accurate results for multicast data transfer. Using a convenient simulator, the optimized transport protocol can be *synthesized* for a given media application, satisfying the requirements. This means that by means of a suitable mathematical method, an optimal point in the hyperspace of the protocol parameters can be found.

The optimization process should deal with the dependencies between protocol parameters. Most dependencies should be taken into account, but some of them can be omitted. An example of a negligible dependency is the relation between the protocol parameter *Feedback control* and the protocol parameter *Feedback addressee*, where modifying the actual value of the *Feedback control* from *Structure-based* to *Timer-based* could change the *optimal* value of *Feedback addressee* from *Intermediate host* to *Every member*, however, its influence may practically be ignored.

The developed simulator became a helpful tool for the analysis of multicast transport protocol mechanisms (Orosz & Tegze, 2001). By means of this simulator an optimized transport protocol can be synthesized, satisfying the requirements of certain type of media applications.

SIMULATING THE MULTICAST TRANSMISSION

The multicasting capabilities of the software are demonstrated by a distance vector based multicast protocol implemented in the simulator, which is very much like the existing DVMRP protocol (Distance Vector Multicast Routing Protocol). The concept of this protocol is similar to that of RIP (Routing Information Protocol) unicast routing protocol, which is one of the most common interior gateway protocols. This DVMRP like protocol determinates the multicast routing table by means of a flooding algorithm. Routers send *advertising packets* to the neighbor routers periodically to propagate routing information about them. Receiving such a packet, a router forwards the packet through all of its interfaces to the next router, except the receiving interface, and increments the hop counter field in the packet header. If the hop counter and source address of the received packet indicates a new route or a route better than the previous one, then the router adds it to its multicast routing table or modifies an existing table entry. To avoid infinite circulation of advertising packets, a TTL (Time To Live) like mechanism is used. This means that the source of the packet sets the TTL field of the packet to a positive integer. Each time the packet gets forwarded, this

Table 1. The selected set of the protocol parameters

Protocol parameter	Values
Flow control	Window-based, Rate-based, Multigroup multicast, Receiver give-up, None
Data accuracy	Reliable, Atomic, Non-reliable
Feedback addressee	Original source, Intermediate host, Every member, None
State control	Sender-based, Receiver-based, Shared, None
Feedback control	Structure-based, Timer-based, Representatives-based, Rate-based, None
Way of sending repair	Unicast, Multicast, None
Scope of repair	Global, Global to secondary group, Global to individual members, Local, None
Session membership control	Explicit, Implicit, None