Decoupling QOS Guarantees and Connection Establishment in Communication Networks.

Dinesh C. Verma

Philips Research
345 Scarborough Road
Briarcliff Manor, NY -10510
Phone: (914)-945-6079
FAX: (914)-945-6375
Email: `dcv@philabs.research.philips.com`

Abstract

Quality of Service guarantees are necessary for efficient network communication for distributed multimedia applications. However, most proposed QOS schemes couple resource allocation and QOS guarantees with connection establishment, and therefore fail to perform efficiently for applications like WWW browsing which establish short-lived connections frequently. In this paper, we argue that resource allocation be separated from connection establishment, and that resources be allocated in the network on an abstraction that is separate from that of a connection. We explore the issues in decoupling connection establishment and QOS reservation, and propose a new network architecture that addresses some of those issues.

1. Introduction

The need to provide quality of service for continuous-media and real-time applications is well recognized by the networking research community[1]. Towards this purpose, several approaches have been proposed for network protocols and architectures that support the notion of guaranteed performance or Quality of Service in packet-switched networks. Some of the most notable ones are RSVP[2], Tenet real-time channels[3] and ST-II[4]. Others include the Heidelberg Transport System [5] Berkom MMTS [6]and IBM NBBS architecture[7]. The ATM Forum specifications [8] provide support for QOS and the integrated services group in IETF is examining various ways to provide QOS and performance guarantees.

While there are differences in the way the schemes operate(e.g. RSVP is receiver-oriented while the ATM Forum is sender-oriented), the schemes follow very similar paradigms. Resources are reserved on a per-session basis at connection establishment time and released when the connection is taken down. The schemes operate by establishing network layer QOS-guaranteed connections (e.g. RSVP flows), and usually one transport-layer connection is mapped to a single network-layer connection. The schemes would work well if most of the connections in the network have a life that is long compared to the connection establishment time. Furthermore, resource allocation is usually a slow process, and these schemes work best when connection establishment rate is relatively slow.

For the most common application in the Internet, namely WWW browsing, this characteristic is not true. The browsers typically establish several short-lived sessions as the user follows chains in the hypertext. The establishment of connections is fairly frequent, since a single web-page access can generate several TCP connection establishment requests. Thus, most of the proposed schemes would fail to perform well for the current Internet work-load. Of course, there are other application areas where each of the schemes would be appropriate.

In this paper, we propose that quality of service in the network be decoupled from the connection in order to solve the mismatch between the needs of the applications and the solutions provided by the networks.We first consider two obvious methods that can be used to address this issue, namely fast-establishment and multiplexing, and examine their advantages and disadvantages. Then, we examine the issue of network resource wastage in connection establishment. Finally, we outline a packet-bus approach which can be used to solve some of the problems in QOS guarantees to frequently established short-lived connections.

2. Fast Connection Establishment

Fast (or optimistic) connection establishment attempts to address the slowness of connection establishment by piggy-backing the connection establishment request on the first data packet on the connection. Thus, the data that needs to be transmitted does not need to wait for the connection establishment hand-shake to complete. The scheme assumes that the resources would be allocated with a high probability of success, and thus the packets in the network would most frequently be able to get across.

While fast establishment is appropriate in networks where the resource allocation process during connection establishment can be performed quickly, it loses its advantage in router/switch architectures where the paths taken by a data packet and by a connection establishment request are different. In order to streamline packet forwarding, several routers and switches perform basic packet forwarding in adapters attached to network links, and off-load more complex functionalities to an attached processor. In such architectures, fast establishment has the drawback that the subsequent data packets may easily over-take the first data/connection-establishment packet.

Another disadvantage of fast connection establishment is in its optimistic assumption. If a large percentage of packets are able to obtain the desired quality of service on the fly, resource reservation itself does not add much value to the basic packet forwarding function. Under the conditions where optimistic connection establishment succeeds over 99% of the time, one can justifiably argue that the network forego any kind of resource reservation, and have the network operate on a best-effort basis only.

3. Multiplexing

An alternate solution to the relative slow resource reservation architecture is the use of multiplexing. Instead of establishing one QOS-guaranteed network connection for every transport connection establishment request, only one (or a small number of) network connections is established. Several of the transport connections originating on the node can be multiplexed on to the same network-layer connection, and established only by hand-shaking between the end-points of the connections. This increases the life of a network connection and decreases the effective network connection establishment rate. The approach, however, assumes that the requirements of the different transport connections are similar to that of the connection that are being multiplexed upon. Furthermore, the multiplexing can only be done when the source and the destinations of the transport connections are the same as those of the network connections.

While multiplexing multiple transport-level connection does make the network connection establishment rate somewhat manageable, it fragments the available network resources into several semi-permanent connections. The resources reserved for the semi-permanent network level connection can not be reused for other reserved-bandwidth applications, with the result that connections requiring QOS guarantees may be refused because one connection has over-allocated its resources, and not using them.

Resources can get wasted in the network due to two primary reasons. In order to obtain performance guarantees to connections, the network nodes (switches or routers) perform an admission control scheme. The admission control scheme tends to make pessimistic assumptions in order to ensure quality of service to accepted connections. This tends to waste the available resources. For almost all admission control algorithms, the wastage is worse when the traffic on the network is bursty. In addition to this external wastage, there is an internal wastage since each connection has to ask for resources that are slightly higher than its actual usage. For example, a session that is expected to generate packets at a rate r must ask for a network connection that guarantees a bandwidth slightly higher than r in order to have finite queueing delays and loss probabilities. Most QOS architectures attempt to reduce wastage by permitting best-effort packets to take up the remaining bandwidth, but the resource is still wasted from the perspective of other connections requiring QOS guarantees.

In order to reduce external wastage of resources, we need to make the traffic on the network smoother and periodic. In order to reduce internal wastage, we should permit a connection to use the resources reserved for another connection when they are not being used by the latter's packets.

Towards this goal of reducing network wastage, we propose a new QOS architecture for communication networks. The architecture postulates two kinds of entities that circulate in the network, the first one being the normal packet (or cell) which are always carried on a best-effort basis, and the second one being a packet-bus which are always carried with a guaranteed QOS.

4. Packet Bus Architecture

A packet bus consists of several *seats*, with a limit on the maximum number of seats. Each seat is a slot of fixed length, and a packet may take one or more seats in a bus depending on its size. A packet-bus has a known path (*bus-route*) in the network, and any packet that originates at a node along the path can *catch the bus* along the path and *get-down* from the bus on any downstream node. Of course, the packet needs to *purchase a ticket* for the bus. Packet buses are scheduled in advance, and *operate* at fixed times.

The entire network contains several packet-buses that are pre-scheduled and operate at regular intervals. Because of the regularity and predictability of the packet-buses, nodes and switches in the network can predict in advance when a bus would arrive, and can provide a guaranteed quality of service to all packets in the bus. In addition to scheduled packet-buses, the network may contain connection-less traffic or connection-oriented traffic using any of the reservation schemes mentioned above.

When a packet has to be transmitted from a source to the destination, the sending node has to decide on the bus that it wants the packet to take, purchase a ticket for the bus, wait for the bus to arrive, and send the packet on the bus. At each node on the bus route, the node examines if there are any packets that are destined for it, and takes them out. A packet which needs performance guarantees needs to purchase the ticket in advance so that it is guaranteed a spot on the bus. A best-effort packet can try to purchase the ticket on the bus, i.e. take the bus if there is space in it.

Since the time when a packet bus would arrive is known in advance, and its capacity is known as well, a node can easily determine whether it has enough resources to meet a schedule. Any of the various admission control tests used in conjunction with the schemes of [1] -[8] can be used for this purpose. In order to provide QOS in the network, each node has to allocate enough space for the packets in the bus, and decide on an upper bound on the delay in processing packets in the bus. Thus, each bus would have a designated *stopping time* at a node.

A packet needing QOS guarantees has to decide on the bus it desires to catch and obtain a ticket on it. Once the packet has the ticket, it is guaranteed to have a QOS similar to that of the bus. In some cases, when a good bus is too late to be useful, the packet may opt to go for a best-effort service on its own.

The implementation of packet-buses requires three components (a) A *ticket agent* who issues tickets to the packets (b) A *bus scheduler* who decides which buses to run and when to run them, and (c) A *boarding agent* at each station which decides which packets to embark/disembark from the bus. Each of these problems can be solved in either a centralized or a distributed manner. Additionally, we need to define headers which would define the contents of a packet bus.

Packet buses provide an interesting mechanism for providing QOS in a connection-less network. For this work-shop, we take the position that an approach like the packet-bus, which decouples connection establishment and resource reservation, is a better way to build a network that offers QOS guarantees.

References

[1] A Vogel, B. Kerherve, G. von Bochmann and J. Gecsei, “Distributed Multimedia and QOS: A Survey” IEEE Multimedia, Summer 1995, pp. 10-18.

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, “RSVP: A New Resource Reservation Protocol”, IEEE Network, September 1993, pp. 9-18.

[3] D. Ferrari, A. Banerjea and H. Zhang, “Network Support for Multimedia”, Tech. Report TR-92-072, The International Computer Science Institute, Berkeley, California, 1992.

[4] C. Topolcic, “Experimental Internet Stream Protocol Version 2 ST-II”, Internet RFC 1190, 1990.

[5] L. Delgrossi, R.G. Herrtwich and F.O. Hoffmann, “An implementation of ST-II for the Heidelberg Transport System”, Proc. IEEE Globecom, 1992.

[6] S. Boecking et. al, “The Berkom Multimedia Transport System”, Proc. IS&T/SPIE Symposium on Electronic Imaging, Science & Technology, Workshop on High-Speed Networking and Multimedia Computing, 1994, pp. 256-265.

[7] C. Marin, C. Immanuel, P. Chimento and I. Gopal, “Overview of the NBBS Architecture”, IBM Systems Journal, vol. 34, no 4, December 1995.

[8] The ATM Forum “Traffic Management Specifications”, draft version dated February 1996.