JOURNAL OF INFORMATION, KNOWLEDGE AND RESEARCH IN COMPUTER ENGINEERING

PACKET LOSS CONTROL USING TOKENS AT THE NETWORK EDGE

[1] K.K. NIKHIL, [2] G. SUNIL SANTHOSH KUMAR

[1,2] ASST.PROFESSOR,CSE DEPARTMENT, AVR & SVR CET

ABSTRACT — *Packet loss is taboo; to an Internet architect, it immediately signifies an inefficient design likely to exhibit instability and poor performance. In this paper, we argue that such an implication is not fundamental. In particular, there exist design points that provide many desirable properties including near optimal performance while suffering high loss rates. We focus specifically on congestion control, where researchers have long clung to the belief that loss avoidance is central to high throughput. A protocol that supports the sharing of resources that exist in different packet switching networks is presented. The protocol provides for variation in individual network packet sizes, transmission failures, sequencing, flow control, end-to-end error checking, and the creation and destruction of logical process-to-process connections. Some implementation issues are considered, and problems such as internetwork routing, accounting, and timeouts are exposed. initial TCP congestion control algorithm, the entire tradition of end-to-end congestion control has attempted to optimize network performance by tempering transmission rates in response to loss. We argue that by removing the unnecessary yoke of loss avoidance from congestion control protocols, by using Random Early Detection (RED)Detect incipient congestion. They can become less complex yet simultaneously more efficient, stable, and robust.*

Keywords: TCP, Tokens, Network, Congestion Control Algorithm, Addressing, Formatting, Buffering, Sequencing, Flow Control, Error Control, Qos, Random Early Detection (RED).

I.INTRODUCTION

There are a number of very good reasons to avoid loss in today's networks. Many of these stem from the fact that loss is often a symptom of overflowing router buffers in the network, which can also lead to high latency, jitter, and poor fairness. In the last few years considerable effort has been expended on the design and implementation of packet switching networks [1,2]A principle reason for developing such networks has been to facilitate the sharing of computer resources.

In this paper, we study whether the benefits of a network architecture that embraces rather than avoids widespread packet loss outweigh the potential loss in efficiency. We propose an alternative approach to Internet congestion control called decongestion control.

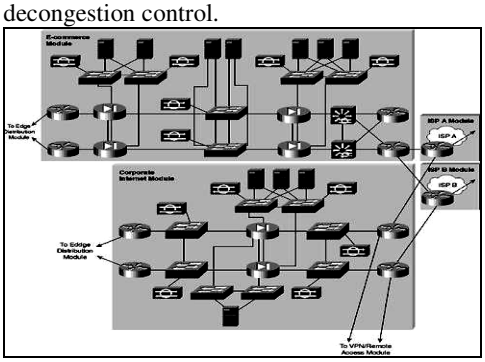


Fig1: packet switching communications at network Edge

In a departure from traditional approaches, end hosts strive to transmit packets faster than the network can deliver them, leveraging end-to-end erasure coding and in-network fairness enforcement. In this paper we present a protocol design and philosophy that supports the sharing of resources that exist in different packet switching networks. After a brief introduction to internetwork protocol issues, we describe the function of a GATEWAY as an interface between networks and discuss its role in the protocol. [2,3,4]We then consider the various details of the protocol, including addressing, formatting, buffering, sequencing, flow control, error control, and so forth. A typical packet switching network is composed of a set of computer resources called HOSTS, a set of one or more *packet switches*, and a collection of communication media that interconnect the packet switches.The ensemble of packet switches and communication media is called the *packet switching subnet* as shown in Fig. 1. In a typical packet switching subnet, data of a fixed maximum size are accepted from a source HOST, together with a formatted destination address which is used to route the data in a store and forward fashion.

II.IMPLEMENTATION:

Such an approach (sometimes referred to as a firehose) is often dismissed in the literature due to the potential for congestion collapse a condition in which the network fig2 is saturated with packets but total end-to-end goodput is low. However, congestion collapse occurs only under two conditions: if receivers are unable to deal with high loss[5,6] (so-

called classical congestion collapse), or if the network topology is such that packet drops occur deep in the network, thereby consuming network resources that could be fruitfully consumed by other flows [6,7,8]. The first concern can be addressed by applying efficient erasure coding [8,9]. It is unknown whether the second condition arises frequently in practice; it occurs rarely in the backbone topologies we study.

2.1Efficiency. Sending packets faster than the bottleneck capacity ensures utilization of all available network resources between source and destination. With appropriate use of erasure codes, almost all delivered packets will be useful.

2.2Simplicity. Because coding renders packet drops (and reordering) inconsequential, it may be possible to simplify the design of routers and dispense with the need for expensive, power-hungry fast line-card memory.

2.3Stability. Decongestion transforms a sender's main task from adjusting transmission rate to ensuring an appropriate encoding. Unlike the former, however, one can design a protocol that adjusts the latter without impacting other flows.

2.4Robustness. Existing congestion control protocols are susceptible to a variety of sender misbehaviors, many of which cannot be mitigated by router fairness enforcement. Because end points are already forced to cope with high levels of loss and reordering in steady state, decongestion is inherently more tolerant. The transmit time for this data is usually dependent upon internal network parameters such as communication media data rates, buffering and signalling strategies, routeing, propagation delays, etc.

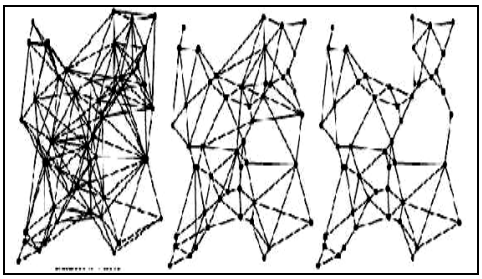


Fig2:Typical network switching circuit

Status information, routing, fault detection, and isolation are typically different in each network. We obviously want to allow conversion between packet switching strategies at the interface, to permit interconnection of existing and planned networks.

III.DECONGESTION CONTROL IS OPTIMISTIC:

senders always attempt to over-drive network links. Should available capacity increase at any router due to, for example, the completion of a flow, the remaining flows instantaneously take advantage of the freed link resources. To translate increased throughput into increased goodput, senders encode flows using an erasure coding scheme appropriate for the path loss rate experienced by the receiver.

IV.TRAFFIC AND RESOURCE MANAGEMENT:

Resources statistically shared: In computing, a shared resource or network share is a device or piece of information on a computer that can be remotely accessed from another computer, typically via a local area network or an enterprise Intranet, transparently as if it[10,11]were a resource in the local machine. Examples are shared file access (also known as disk sharing and folder sharing), shared printer access (printer sharing), shared scanner access, etc. The shared resource is called a shared disk (also known as mounted disk), shared drive volume, shared folder, shared file, shared document, shared printer or shared scanner. The term file sharing traditionally means shared file access, especially in the context of operating systems and LAN and Intranet services, for example in Microsoft Windows documentation

4.1.OVERLOAD CAUSES CONGESTION: it causes 2 options one is packet delayed or dropped And second is application performance suffer with Local vs. network wide fig3 and Transient vs. persistent.And Challenges are high resource utilization and high application performance.

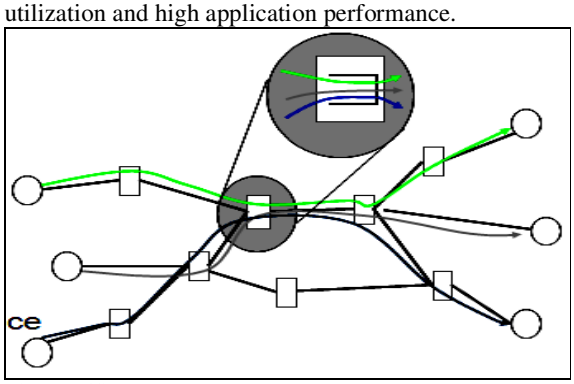


Fig3:traffic resource management

$$\sum Demand_i(t) > \mathrm{Re}\,source(t)$$

- Increase resources
- install new links, faster routers
- capacity planning, provisioning, traffic engineering
- happen at longer timescale
- Reduce or delay demand
- Reactive approach: encourage everyone to reduce or delay demand
- Reservation approach: some requests will be rejected by the network

V.CONGESTION CONTROL IN INTERNET: End-system-only solution (TCP) in fig4

- dynamically estimates network state
- packet loss signals congestion
- reduces transmission rate in presence of congestion
- routers play little role

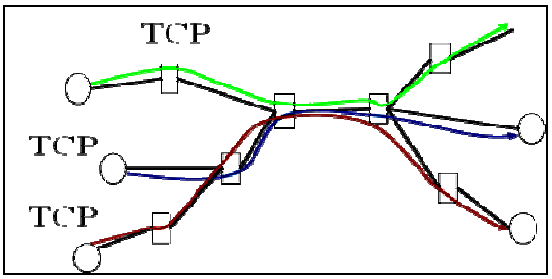


Fig4:congestion control in network

In Fig. 5 we illustrate three individual networks labelled *A*, *B*, and *C* which are joined by In practice, a GATEWAY between two networks may be composed of two halves, each associated with its own network. It is possible to implement [11.12]each half of a GATEWAY so it need only embed internetwork packets in local packet format or extract them. We propose that the GATEWAY handle internetwork packets in a standard format, but we are not proposing any particular transmission procedure between GATEWAY halves. Let us now trace the flow of data through the interconnected networks. We assume a packet of data from process *X* enters network *A* destined for process *Y* in network *C*. The address of *Y* is initially specified by process *X* and the address of GATEWAY *M* is derived from the address of process *Y*. We make no attempt to specify whether the choice of GATEWAY is made by process *X*, its HOST, or one of the packet switches in network *A*. The packet traverses network *A* until it reaches GATEWAY *M*. At the GATEWAY, the packet is reformatted to meet the requirements of network *B*, account is taken of this unit of flow between *A* and *B*, and the GATEWAY delivers the packet to network *B*. Again the derivation of the next GATEWAY address is accomplished based on the address of the destination *Y*. In this case, GATEWAY *N* is the next one. The packet traverses network *B* until it finally reaches GATEWAY *N* where it is formatted to meet the requirements of network *C*. Account is again taken of this unit of flow between networks *B* and *C*. Upon entering network *C*, the packet is routed to the HOST in which process *Y* resides and there it is delivered to its ultimate destination.

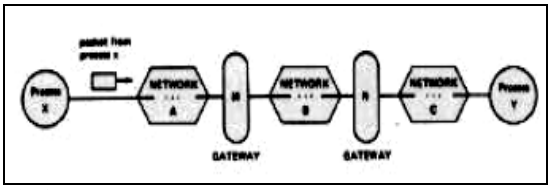


Fig5:network connected by 2 gate ways

1) If a maximum permitted packet size is specified then it becomes impossible to completely
isolate the internal packet size parameters of one network from the internal packet size parameters of all other networks.

2) It would be very difficult to increase the maximum permitted packet size in response to new technology (e.g. large memory systems, higher data rate communication facilities, etc.) since this would require the agreement and then implementation by all participating networks.

3) Associative addressing and packet encryption may require the size of a particular packet to expand during transit for incorporation of new information. Provision for fragmentation (regardless of where it is performed) permits packet size variations to be handled on an individual network basis without global administration and also permits HOSTS and processes to be insulated from changes in the packet sizes permitted in any networks through which their data must pass

VI.RTT FEEDBACK CONTROL:

A data stream (e.g. a continuously generated bit string) can be represented as a sequence of finite length messages. Within a HOST we assume that existence of a transmission control program (TCP) which handles the transmission and acceptance of messages on behalf of the processes it serves. The TCP is in turn served by one or more packet switches[11] connected to the HOST in which the TCP resides. Processes that want to communicate present messages to the TCP for transmission, and TCP's deliver incoming messages to the appropriate destination processes. We allow the TCP to break up messages into segments because the destination may restrict the amount of data that may arrive, because the local network may limit the maximum transmission size, or because the TCP may need to share its resources among many processes concurrently With RTT feed back control as shown in below fig6.

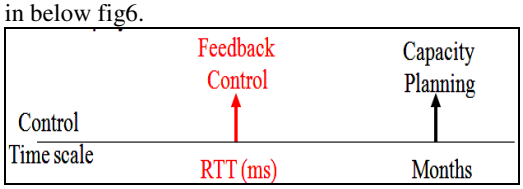


Fig6:RTT feed back control

6.1THE IMPACT OF LOSS:

There are two distinct impacts of packet loss, direct and indirect. Packet loss within a flow directly reduces the throughput delivered to the intended receiver. Indirectly, packet loss may cause inefficient use of upstream network resources: flows other than the one experiencing loss may have been able to put the upstream capacity to better use.

6.2ADDRESS FORMATS:

The selection of address formats in fig7 is a problem between networks because the local network addresses of TCP's may vary substantially in format and size.

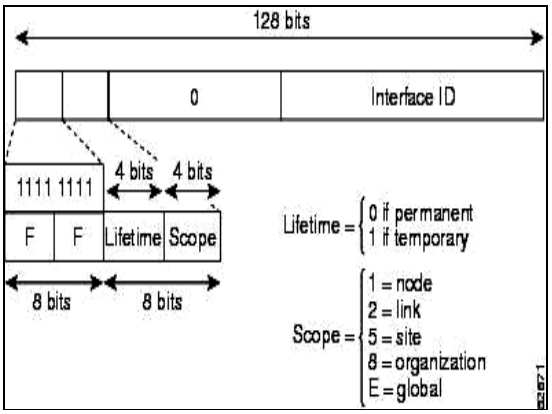


Fig7:address format in TCP

A uniform internetwork TCP address space, understood by each GATEWAY and TCP, is essential to routing and delivery of internetwork packets. TCP addressing is intimately bound up in routeing issues, since a HOST or GATEWAY must choose a suitable destination HOST or GATEWAY for an outgoing internetwork packet.

VII.FLOW CONTROL:

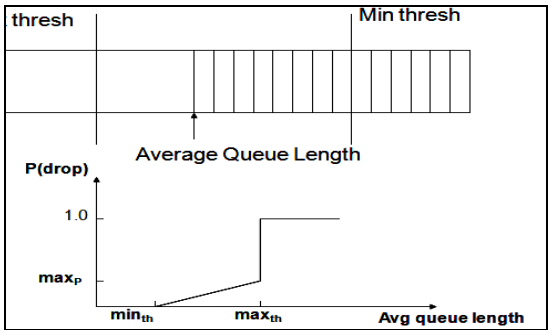
Normally, we would expect the sender to abide by the window limitation. Expansion of the window by the receiver merely allows more data to be accepted. For the receiving HOST with a small amount of buffer space, a strategy of discarding all packets whose sequence numbers do not coincide with the current left edge of the window is probably necessary, but it will incur the expense of extra delay and overhead for retransmission.Every segment[12]that arrives at the destination TCP is ultimately acknowlegded by returning the sequence number of the next segment which must be passed to the process (it may not yet have arrived). Earlier we described the use of a sequence number space and window to aid in duplicate detection. Acknowledgments are carried in the process header and along with them there is provision for a "suggested window" which the receiver can use to control the flow of data from the sender. This is intended to be the main component of the process flow control mechanism..

VIII.RANDOM EARLY DETECTION (RED)

1. Detect incipient congestion
2. Assume hosts respond to lost packets
3. Avoid window synchronization

a. Randomly mark packets

4. Avoid bias against bursty traffic



RED ALGORITHM:in fig8 consists

1.Maintain running average of queue length

2.If avg < min_{th} do nothing

3.Low queuing, send packets through

4.If avg > max_{th}, drop packet

5.Protection from misbehaving sources

6.Else mark packet in a manner proportional to queue length

7.Notify sources of incipient congestion

Much of the thinking about process-to-process communication in paket switched networks has been influenced by the ubiquitous telephone system. The HOST-HOST protocol for the ARPANET deals explicitly with the opening and closing of simplex connections between processes [13],[14]. Evidence has been presented that message-based "connectionfree" protocols can be constructed [15], and this leads us to carefully examine the notion of a connection. The term *connection* has a wide variety of meanings. It can refer to a physical or logical path between two entities, it can refer to the flow over the path, it can inferentially refer to an action associated with the setting up of a path, or it can refer to an association between two or more entities, with or without regard to any path between them. In this paper, we do not explicitly reject the term connection, since it is in such widespread use, and does connote a meaningful relation, but consider it exclusively in the sense of an association between two or more entities without regard to a path. To be more precise about our intent, we shall define the relationship between two or more ports that are in communication, or are prepared to communicate to be an *association*. Ports that are associated with each other are called *associates*.

IX.CONCLUSIONS: Improve TCP and Stay with end-point only architecture Enhance routers to help TCP and Random Early Discard with Enhance routers to control traffic and Rate limiting and Fair Queueing and Provide QoS by limiting congestion.We have discussed some fundamental issues related to the interconnection of packet switching networks. In particular, we have described a simple but very powerful and flexible protocol which provides for variation in individual network packet sizes, transmission failures, sequencing, flow control, and the creation and destruction of processto-process associations. We have considered some of the implementation issues that arise and found that

the proposed protocol is implementable by HOST'S of widely varying capacity. The next important step is to produce a detailed specification of the protocol so that some initial experiments with it can be performed. These experiments are needed to determine some of the operational parameters of the proposed protocol.

REFERENCES

[1] G. Appenzeller, N. McKeown, J. Sommers, and P. Barford, "Recent Results on Sizing Router Buffers," in Proceedings of the Network Systems Design Conference, Oct. 2004

[2] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: Routers with very small buffers," ACM/SIGCOMM Computer Communication Review, vol. 35, pp. 83-90, July 2005.

[3] L. Zhang, S. Shenker, and D. Clark, "Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic," in Proceeedings of ACM SIGCOMM, pp. 133–147, Sept. 1991. [3] F. R. E. Dell, "Features of a proposed synchronous data network," in Proc. 2nd Symp. Problems in the Optimization of Data Communications Systems, 1971, pp. 50—57.

[4] R. A. Scantlebury and P. T. Wilkinson, "The design of a switching system to allow remote access to computer services by other computers and terminal devices," in Proc. 2nd Symp. Problems in the Optimization of Data Communications Systems, 1971, pp. 160-167.

[5] D. L. A. Barber, "The European computer network project," in Computer Communications: Impacts and Implications, S. Winkler, Ed. Washington , D.C., 1972, pp. 192-200.

[6] R. Despres, "A packet switching network with graceful saturated operation," in Computer rkshop on Quality of Service (IWQoS), June 2005.

Communications: Impacts and Implications, S. Winkler, Ed. Washington, D.C., 1972, pp. 345-351.

[7] R. E. Kahn and W. R. Crowther, "Flow control in a resource-shaping computer network," *IEEE Trans. Commun.,* vol. COM-20, pp. 539-546, June 1972.

[8] J. F. Chambon, M. Elie, J. Le Bihan, G. LeLann, and H. Zimmerman, "Functional specification of transmission station in the CYCLADES network. STST protocol" (in French), I.R.I.A. Tech. Rep. SCH502.3, May 1973.

[9] S. Carr, S. Crocker, and V. Cerf, "HOST-HOST Communication Protocol In the ARPA Network," in *Spring Joint Computer Conf.*, *AFIPS Conf. Proc.*, vol. 36. Montvale, N.J.: AFIPS Press, 1970, pp. 589-597.

[10] A. McKenzie, "HOST/HOST protocol for the ARPA network," in *Current Network Protocols*, Network Information Cen., Menlo Park, Calif., NIC 8246, Jan. 1972.

[11] L. Pouzin, "Address format in Mitranet," NIC 14497, INWG 20, Jan. 1973.

[12] D. Walden, "A system for interprocess communication in a resource sharing computer network," Commun. Ass. Comput. Mach., vol. 15, pp. 221-230, Apr. 1972.

[13] D. Katabi, M. Handley, and C. Rohrs, "Internet congestion control for future high bandwidth-delay product environments," in Proceedings of ACM SIGCOMM, Aug. 2002.

[14] S. Kandula, D. Katabi, B. Davie and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in Proceedings of ACM SIGCOMM 2005, Aug. 2005.

[15] Y. Su and T. Gross, "WXCP: Explicit congestion control for wireless multi-hop networks," in Proceedings of the 13th International Wo