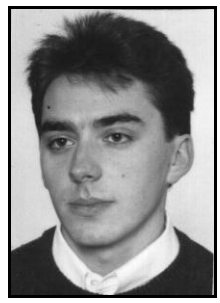
Jerzy WECHTA
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

First Packet Queuing (FPQ) distributed admission control scheme

Dr inż. Jerzy WECHTA



The author of this paper received his PhD degree and the MSc (with distinction) degree from the University of Wales Swansea, UK as well as MEng degree from the Technical University of Zielona Góra, Poland. He has gained his professional experience over the period of six years working as a programmer, group and project manager in a commercial company (ADB, Zielona Góra). Currently he is employed as teaching assistant at the University of Zielona Góra, Poland, carrying out his research.

e-mail: j.wechta@iie.uz.zgora.pl

Abstract

The paper reports on the simulation based research in the area of the data packet admission control in the area of packet based networks (such as Internet). The research results presented show the performance of the new mechanism de-signed by the author. The proposed scheme proves to be beneficial in the area of the problems typical for TCP/IP traffic (congestion, delays, dropped packet, retransmissions, wasted bandwidth). The work presented in this paper has been sponsored by British Telecom and was used as the core material for the patent application by British Telecom.

Keywords: flow, congestion and admission control, FPQ

First Packet Queuing – rozproszony system kontroli admisji

Streszczenie

Artykuł opisuje wyniki badań w obszarze admisji pakietów danych w sieciach pakietowych takich jak Internet. Wyniki badań uzyskanych na drodze symulacji pozwalają ocenić skuteczność nowego mechanizmu wymyślonego przez autora. Zaproponowane rozwiązanie okazało się w efekcie badań bardzo pomocne w walce z objawami przeciążenia w sieci takimi jak opóźnienia pakietów, porzucanie pakietów, retransmisje, zmarnotrawiona przepustowość. Badania opisane w artykule były sponsorowane przez British Telecom i zostały użyte jako podstawowy materiał do wniosku patentowego zgłoszonego przez British Telecom.

Słowa kluczowe: zarządzanie przepływem, przeciążenie sieci i admisja w sieciach pakietowych, FPQ

1. Introduction

Almost 100% of the traffic in today's Internet is generated by the TCP/IP family of protocols (FTP, HTTP, SMTP, POP3, IMAP). Almost 100% of the computer applications, working over the Internet (such as web browsing, file transfers, communicators and emails) use the Trnasmission Control Protocol or TCP.

Much has been said in the area of fighting the congestion in the Internet. Congestion in simple words means that the network is offered much higher traffic than it can actually handle. As a result the data packets are delayed or even dropped, which results in the data packet retransmissions, packet reordering etc [1].

Also due to the offen neglected effect of the packets from the same flow traveling together and as many as 8, 16 and more packet can be dropped in a row from the same flow or session– an interactive user is likely to drop the connections and restart new ones. All the above causes the wastage of a valuable bandwidth.

Additionally in the era of network telephony, network radio transmissions and network TV, the issues of guaranteed bandwidth, controlled delay and the number of dropped packets (ofen described in a form of the required QoS) becomes even more burning [2].

The main directions, in which the congestion related research is carried out are the data flow, congestion and admission control. Currently deployed solutions for handling the congestion can be shortly listed as follows. First one is using the connection based networks (e.g. ATM). This solution is expensive and not well scalable in the ever growing Internet. Second solution is related to traffic profiling, shaping and creating reservations in connectionless networks (e.g. RSVP) [3]. However, Trafic profiling for the TCP flows is very difficult and additionally shaping can be done at the edges of the network where the no way of knowing what is the current situation on any congested links. Third solution is Differentiated Services (DS) [3], which is most promising solution, but requires the obedience of certain rules by all of the bodies creating the Internet, which at the moment is impossible to be forced on.

Although there also exist some packet marking methods (e.g. TCP source quench signal, RED used for early connection notification or ECN) but this are not considered fast and strong enough solutions.

Nevertheless the most popular way of approaching the congestion is dropping packets when the buffers get filled in front of a congested link (so called DROP Tail [4]). Even if some clever mechanisms have been proposed (to prevent dropping many packets from the same data connection or flow) and actually deployed in the network devices (e.g. random early detection or RED [Fair RED or FRED , Weighted RED or WRED and more) these use dropping packets as a way of conveying the information back to the source that the network is busy and the source has to slow down.

However, although such schemes produce some free buffer space in the FIFO queues (and hence packet buffers) associated with a congested link, the inevitable effect of this is to compromise the QoS parameters of the affected streams. Also, in applications that involve the retransmission of lost packets by the source end system, the relief obtained by these approaches can only be temporary and is associated with wasting bandwidth.

All of the above solutions try - with various results - to satisfy the two fundamental rules of an efficient network:

- the throughput of any congested link has to be maximized to avoid situation that no packets are ready to be sent on the link which is a bottleneck
- any packet on any link has to experience a minimal delay by minimizing the time it spends in the buffers

Unfortunately these two fundamental rules contradicts each other. The internet itself is a very dynamic collection of various behaviours, reactions, states of resources, changing flows and conditions and hence a method, which can be deployed at the very source of the problems (in this case congestion) exactly at the time when the problem occurs is critical.

The idea of the FPQ becomes easy to understand if one looks at the moving data traffic and the congested point in the network as the entities holding all the necessary information for making an optimal decision. The solution described in the next section shows a new way of thinking about the data traffic traveling through the Internet and proposes a very flexible, easy in implementation and powerful way of approaching the congestion problem by means of controlling the admission of new flows.

2. First Packet Queuing or FPQ

A key premise of the FPQ scheme is that when congestion at a particular outgoing link associated with a network node starts, in

order to avoid the QoS guarantees of all the existing flows that are using that link being compromised, it is essential to stop or delay the admission of any new calls/sessions from using the congested link. The key issue, therefore, is how new calls/sessions are prevented from being admitted to the network whilst still maintaining the CL mode of working.

In practice, irrespective of the operational mode of the network, the first packet associated with each new call/session is always concerned with the source end system requesting the destination end system to be involved in a call/session. Examples include, for a conventional data-only session over the Internet, an IP packet containing a TCP SYN segment or, for a call involving real-time/streaming media such as that associated with a telephone or videophone call, a packet containing a call setup request relating to a signalling protocol. Hence in order to exploit this fact, the first packet relating to each new call/session is marked by the source using, for example, a single bit in the IP packet header.

Then, when a network node processes each packet, it first tests this bit and, if set, it either forwards the (first) packet in the normal way if no congestion is present or, if congestion is present, the packet is either dropped - a technique called first packet dropping (FPD) - or is delayed by placing it in a separate FIFO queue associated with the required outgoing link. This is known as the first-packet queue and the approach, first-packet queueing (FPQ). Packets delayed in this way are then only forwarded if the congestion subsides within a defined time of the packet being queued. Otherwise the packet is discarded.

Clearly, a number of different strategies can be used to determine when a first-packet should be delayed and when a delayed first-packet should be transmitted. For example, the queue manager associated with each outgoing link (in each network node) can keep a record of the number of packets that are queued for the link. Then, only if the queue length is below a defined threshold is a first-packet placed in the queue for forwarding, otherwise it is either discarded - FPD - or is delayed by placing it in the first-packet queue - FPQ.

Similarly, if the number of queued packets for a link falls below a second threshold then a delayed first-packet is transferred from the first-packet queue to the link output queue for forwarding. Also, if a first-packet is queued for longer than a defined time-out interval, then it may be discarded. In this case, the lack of a call accepted or session-acknowledgement packet (relating to the discarded first-packet) being received by the source end system will result in the initiating protocol timing out and sending a new session-request first-packet. A typical delay, for example, is in the order of 6s with TCP which, for this type of traffic, may be sufficient for congestion to ease and to allow the next first-packet to be successful.

It should be stressed that the proposed scheme was devised with TCP flows in mind, mainly because the number of packets sent by TCP is controlled by a congestion window. Additionally, TCP flow control does not allow a second packet to be sent before the first one is delivered and its delivery is confirmed by the arrival of a corresponding ACK packet.

The idea of admission control for TCP is relatively new one. As far as the existing admission control schemes are concerned most of them address UDP based constant rate or adaptive rate flows. This can be easily explained by the fact that it is much easier to estimate the data rate of such a flow, and hence provide efficient bandwidth management for real time multimedia flows. UDP-based flows fit well into existing and well researched types of admission control, namely measurement based and parameter -based [5].

The idea of rejecting new flows when buffers start filling up is related to the observation that there is always a bottleneck link in a network and the majority of the data packets sent by all sources are likely to be queued in buffers located in front of this bottleneck link

Stopping or delaying the first-packet of a flow is much more powerful and effective because it stops all data packets from the affected flow and hence represents a much more effective way of avoiding data loses due to buffer overflow and relieving congestion. Additionally, first-packets (SYN) do not usually carry data, which means firstly, that less buffer space is needed to queue such a packet and secondly if they are dropped the associated bandwidth wastage is not so severe.

Due to the fact that FPQ is designed to be used within stateless routers the algorithm does not depend on information such as number of flows in progress or number of established connections. It also does not rely on identifying load of any particular flow and does not monitor setting up and tearing down data connections. Additionally, it does not use any kind of explicit signalling.

Depending on how sophisticated the first-packet handling and decision making mechanism is, it would make decisions with respect to first-packets on three levels:

- What to do with arrived first-packet: relay, queue or drop
- When to make a decision: decision based on instantaneous or averaged buffer occupancy (e.g. exponentially weighted moving average)
- Which packet out of packets stored within first-packet queue forward/admit first

The proposed active queue management scheme is based on a common believe that preventive flow control is a better solution than reactive flow control and hence it is claimed that the proposed scheme performs better than currently used algorithms (e.g. Drop-tail and RED).

It should be also stressed that one of the biggest advantages of the proposed scheme lies in the fact that there are no changes necessary to TCP end-to-end flow control, because the retransmission of the dropped first-packet is treated by TCP in the appropriate way. There is also no need to deploy any rate shaping mechanisms (e.g. leaky bucket) for TCP flows.

3. Simulation setup

The research data collected are based on the simulations and use the BONES Designer modeling and simulation software. For the purpose of the research the models of TCP/IP data sources (file transfer), models of routers/switches as well as the first packet queuing modules have been developed and verified.

In order to investigate the potential benefits of using the FPQ scheme many various and extensive simulations have been carried out. The small proportion of results presented here is enough the show the existing relation. With the FPQ scheme, arriving first-packets are transferred either to the link output queue if the current length of the queue is below a defined threshold or to the first-packet queue if the queue length exceeds the threshold. Also, queued packets are either transferred to the link output queue if its length drops below the threshold within a defined time interval or discarded if the end of time interval is reached.

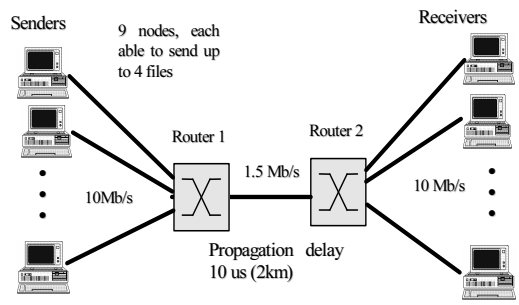


Fig. 1. Analysed topology

In the reported set of simulations a steadily increasing level of offered traffic has been used. Unidirectional traffic and a simple topology are used in order to demonstrate the operation and efficiency of the FPQ scheme. Network related issues such as buffer occupancy, end-to-end delay and throughput are quantified and analysed.

4. Simulation results and discussion

The results of the simulations are summarised in fig. 2 and fig. 3. Fig. 2 relates to Throughput vs. Offered Load and fig. 3 to Mean End-to-end Delay vs. Offered Load. Each figure shows the results from all three simulations. As can be seen in fig. 2 for the Large Buffer case and the Small Buffer case with FPQ enabled, the throughput reaches 100% utilisation and then stays at this level as the offered load keeps increasing. No data packet losses are registered for these two cases.

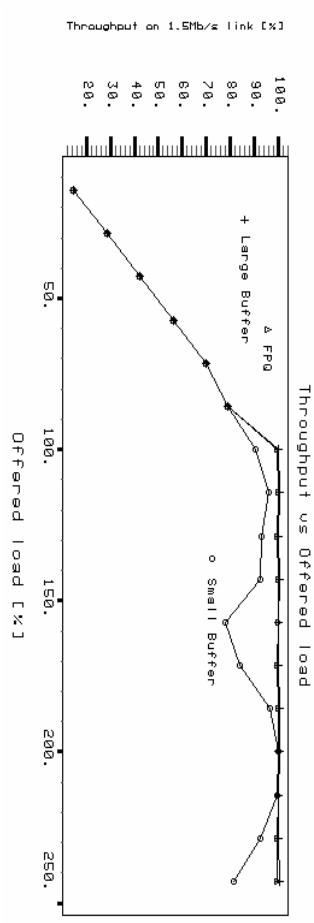


Fig. 2. Throughput vs. Offered load

For the Small Buffer case with FPQ disabled, however, the throughput falls below 100% with the increasing level of offered load. This is caused by an extensive packet dropping. As the offered load increases to 150% the link throughput decreases to below 80%. Since packets are dropped, some TCP sources back off and slow down their transmission and hand over a share of their bandwidth to the unaffected flows. This results in temporary lower buffer utilisation and as new TCP connections are made every second the buffer utilisation and the throughput fluctuate.

As can be seen in fig. 3 the mean delay for the FPQ and Small Buffer cases oscillates around 50ms regardless of the offered load, for the Large Buffer case, however, delays continue increasing with the increasing load and reaches a value of 600ms.

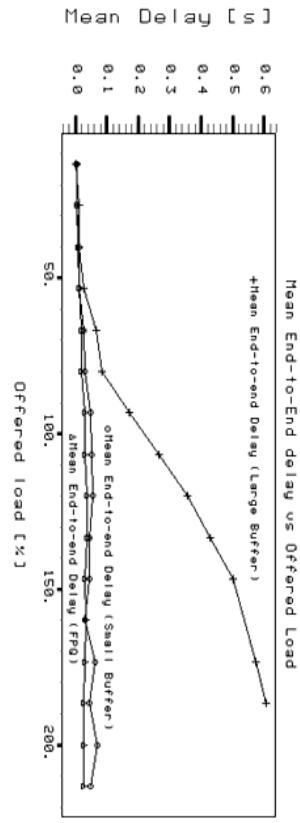


Fig. 3. Mean End-to-end delay vs. Offered load

5. Conclusions and summary

As show the simulation results along with the result discussion the FPQ scheme satisfies the two main rules for the effective network utilization. And hence represents a very interesting family of the new breed of solutions, which can fight the congestion.

The discussion of the wide range of techniques of making decisions on which call/sessions to admit and which reject/queue (e.g. file size, number of sessions from the same user) as well as which calls/sessions to admit first after they have spend some time (e.g. on FIFO bases) and at what speed to release the new flows (e.g. 1 call/session per RTT) exceeds the scope of this paper.

6. Bibliography

[1] Wechta J: Integration of Flow and Congestion Control for Improved Performance of 802.3 Switches (accepted PHD thesis), University of Wales, Swansea, UK, February 2000

[2] Wechta J, Fricker M and Halsall F:Hop-by-hop flow control as a method to improve QoS in 802.3 LANs, International Workshop on Quality of Service '99, UK, 31 May-4 June 1999.

[3] Bernet Y, Yavatkar R, : A Framework for Use of RSVP with Diff-serv Networks, Internet draft: <draft-ietf-diffserv-rsvp-00.txt>, June 1998.

[4] Floyd S and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control", Technical Report, February 1997, URL "ftp://ftp.ee.lbl.gov/ papers/collapse.ps".

[5] Jamin S, Shenker S, and Danzig P :Comparison of Measurement-Based Control Algorithms for Controlled-Load Service, IEEE ACM Transactions on Networking, Vol. 5, No. 1, February 1997, pp. 56-70.

Artykuł recenzowany