NLS needs 20 multiplications and 16 additions. It is clear that, from this perspective, NLS is the better method.

Numerical example: To compare the NLS method to the PCAR method, 200,000 data sets, 100,000 for each bit, were generated for different signal-to-noise ratios (SNRs). The SNR is, in our case, defined as $10\log_{10}(\sigma^2/\sigma^2)$ where $A$ is the amplitude of the sinusoidal signal and $\sigma^2$ is the noise variance. The data sets consisted of four samples each (which is the data available for each bit), generated by embedding a sinusoid with unit amplitude and random phase (uniformly distributed between $-\pi/2$ and $\pi/2$) in white Gaussian noise.

![Fig. 2 Probability of detection of bit 1 for caller ID example](image)

The results are presented in Figs. 1 and 2 where the estimated probability of detection is plotted as a function of the SNR for both PCAR and NLS. As can be seen from Figs. 1 and 2, the NLS method performs better than the PCAR method in the detection of bit 0 while it performs worse in detecting bit 1, yet the difference in performance between the two methods is small. Also, it can be seen that the PCAR method is biased in the sense that for low SNR bit 1 is preferred over bit 0. In addition, it should be noted that the sum of the probabilities of detection for low SNR is 1, which means that the detection for low SNR does not depend on the signal present.

Conclusions: Since the NLS method is computationally two times simpler than the PCAR method and performs similarly to the PCAR method in the caller ID example, it appears that for this application the NLS method should be preferred to the PCAR method, and a drawback of the PCAR method is that it is biased in that for low SNR bit 1 is preferred over bit 0.

© IEE 2000

Electronics Letters Online No: 20001217

D0I: 10.1049/el:20001217

P. Áhgren and P. Stoica (Department of Systems and Control, Information Technology, Uppsala University, P.O. Box 27, SE-75103 Uppsala, Sweden)

E-mail: pa@syscon.uu.se

References


Performance evaluation of fair packet discarding in network with multiple bottlenecks

S. Chan, M. Zukerman, E.W.M. Wong, K.T. Ko and E. Yeung

Fair packet discarding (FPD) is a mechanism which provides incentives to users for participation in congestion control, such that ATM networks can operate in a more efficient manner. The authors study the performance of FPD in a general network with multiple bottlenecks. By simulations, they show that the interaction between FPD and ATM Forum's explicit-rate flow control enables the packet-level throughput of each connection converging to the max-min fair apportionment in a distributed manner. With the built-in policing capability of FPD, the need of dynamic usage parameter control for ABR service is also eliminated.

Introduction: The available bit rate (ABR) service has been developed by the ATM Forum to carry computer communication traffic in asynchronous transfer mode (ATM). The ATM Forum has selected rate-based control as the flow control scheme for ABR service due to its simplicity [1]. The current specification is largely based on a scheme which enhances the enhanced proportional rate control algorithm (EPRCA) [2]. This scheme makes use of the explicit forward congestion indication (EFCI) bit in the payload type (PT) field of the ATM cell header, and resource management (RM) cells to carry congestion information from end to end. The specific use of RM cells depends on the capabilities of the network, and gives rise to two rate control mechanisms referred to as the basic and optional enhanced mechanisms. In both schemes, the source end system (SES) generates one RM cell for every $N_{rm}$ user cells in the forward direction, and these RM cells are returned by the destination end system (DES) in the backward direction to the source.

While considerable progress has been made towards completion of the ABR service specification [3], there are still some open issues which need to be addressed. One of these issues concerns source policing, which is a mechanism that monitors and enforces the source traffic not to exceed their proper rates. Source policing is needed because rate-based schemes penalize sources that obey the feedback information and reward those that do not [4]. These schemes can achieve the purpose of congestion control only if all users act in a co-operative manner. Even a limited number of unco-operative users can cause congestion collapse. As the source rate changes continuously according to the feedback information from the network, such a dynamic nature of the ABR service makes it very difficult to enforce sources to behave properly.

It was shown in [5] that this problem in a network with only a single bottleneck can be overcome by a cell discarding mechanism, fair packet discarding (FPD) [5] which performs source policing implicitly. FPD assumes that sources have end-to-end flow control mechanisms similar to that specified by the ATM Forum, but only react to congestion at their own will. The essence of FPD is to give sources incentives to actively participate in congestion control by reacting to congestion information. It ensures that each source gets no more than its fair share of throughput during overload by discarding all excessive cells. Therefore, only misbehaving sources will suffer from congestion collapse. Note that, as implied by the name, discarding is always carried out in units of packet from the upper protocol layer to reduce bandwidth wastage in transmitting incomplete packets of cells.

Generally, a network will have multiple bottlenecks. In this Letter, we evaluate the performance of FPD in a general network with multiple bottlenecks by simulations. We show that by simply operating FPD independently in each outgoing link of each node in the network, the interaction between end-to-end flow control and FPD enables the packet-level throughput of each connection converging to the max-min fair apportionment in a distributed manner. With max-min fairness maintained, misbehaving sources will not be able to take advantage of co-operative sources and hence usage parameter control is not necessary. Note that we are concerned with the packet-level throughput, rather than cell-level, because this is the actual throughput experienced by the end users. The next Section of this Letter describes the max-min fairness crite-
tion and its computation in a general network with multiple bottlenecks, and, in the Section which follows, the performance of FPD in such a network is studied.

Max-min in general network: The max-min criterion for such general cases can be expressed as follows [6]: while maximising the total throughput, the bandwidth usage of a controlled source will not be lower than that of any other source sharing the same bottleneck link.

Generally, the fair apportionments according to this criterion can be computed by a number of steps which are bounded by the number of links in the network. In each step, the fair apportionment in each link is calculated independently and bandwidth is allocated to the most tightly controlled sources. This procedure is repeated until every source is allocated with a fair apportionment of the capacity. This will be clarified by the following example.

Fig. 1 Network architecture for Example 1

Consider a network of four nodes labelled A, B, C and D, as shown in Fig. 1 (Example 1). Each link is designated by its end nodes, e.g. the link between node A and node B is called link AB. The capacities of links AB, BC and BD are 60, 30 and 60 Mbit/s, respectively. The peak offered load of sources 1, 2, 3 and 4 are 40, 40, 30 and 20 Mbit/s, respectively. To obtain the globally fair apportionments, we first calculate the local fair apportionments in each link independently, which is shown in Table 1. We find that the tightest bottleneck occurs in link BC. Although source 2 can have 30 Mbit/s in link AB, it can only have 15 Mbit/s in link BC, therefore the final apportionment for source 2 is 15 Mbit/s. Since source 3 shares the same bottleneck with source 2, it is allocated 15 Mbit/s. Next, the size of the problem is reduced by removing sources 2 and 3 from the network and subtracting the allocated bandwidth from the link capacities. Since there is no bottleneck remaining in the network, sources 1 and 4 will get 40 and 20 Mbit/s, respectively.

Table 1: Initial bandwidth allocation for Example 1

<table>
<thead>
<tr>
<th>Link</th>
<th>AB</th>
<th>BC</th>
<th>BD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1</td>
<td>30</td>
<td>-</td>
<td>40</td>
</tr>
<tr>
<td>Source 2</td>
<td>30</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Source 3</td>
<td>-</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Source 4</td>
<td>-</td>
<td>-</td>
<td>20</td>
</tr>
</tbody>
</table>

Achieving max-min fairness: In the previous Section, we showed that when there are multiple bottlenecks in a network, the globally fair apportionments are different from those of local ones. This may give the impression that achieving global fairness requires a central controller, or that every node requires complete information on all traffic streams of other nodes. Nevertheless, we shall show that owing to the interactions with the end-to-end adaptive rate-based flow control mechanism, such fair apportionments can be reached by locally and independently operating FPD in each congested link.

Referring again to Example 1, consider that the SESs employ the explicit rate flow control mechanism specified by the ATM Forum. Assuming the sources begin sending traffic at their peak rate, sources 1 and 2 first experience the bottleneck in link AB. With the operation of FPD in this link, each source will get 30 Mbit/s. This apportionment is also recorded in the ER field of the forward travelling RM cells. So, the offered traffic of source 1 in link BD and that of source 2 in link BC are both equal to 30 Mbit/s. With this reduced offered traffic due to FPD, only link BC is still a bottleneck and sources 2 and 3 will each get 15 Mbit/s. Therefore, the end-to-end bandwidth apportionment of sources 2 and 3 are both 15 Mbit/s. Conversely, sources 1 and 4 are not controlled in link BD, and the end-to-end bandwidth apportionment of sources 1 and 4 are therefore 30 and 20 Mbit/s, respectively.

As the RM cells are travelling towards the destinations, their ER fields will be continuously updated to store the minimal bandwidth allocated to the connection. These values will then be conveyed to their sources by the backward travelling RM cells. Upon receiving these values, the SESs can adjust their CCR accordingly. In this example, the CCRs for sources 1 to 3 are reduced to 30, 15, and 15 Mbit/s, respectively. For source 4, it is not controlled at all and its CCR remains at its peak rate. Note that for source policing purposes, when the backward RM cells goes through the access switch, its ER field will be read. This value is then used by the switch to check against the CCR field of the coming forward RM cells and by FPD to calculate the fair apportionment in that link.

With these new rates, link AB is no longer a bottleneck owing to the total demand being only 45 Mbit/s. In this situation, the spare capacity of 15 Mbit/s can be shared equally between sources 1 and 2. So, the fair apportionment for sources 1 and 2 become 37.5 and 22.5 Mbit/s, respectively. Similarly, the fair apportionments of sources 2 and 3 in link BC are both 15 Mbit/s, and the fair apportionments of sources 1 and 4 are 35 and 25 Mbit/s, respectively. For source 1, the process of feedback and rate changes repeats and eventually its CCR will converge to its fair apportionment. Note that in this example, CCRs of sources 2 and 3 have already converged to their fair rates after a round trip time.

We verify the dynamic behaviour described above by cell level simulations. The congestion threshold is equal to half of the switch buffer size. Each source is assumed to have an infinite supply of data. T is chosen to be 1000 cell time. Various simulations have been run, with the switch buffer size equal to 1500, 2500, 3500, 4500 or 5500 cells, and the packet size equal to either 1500 or 4352 bytes. All simulation results exhibit similar behaviours and here we present only one set of the results, in which buffer size equals to 3500 cells and packet size equal to 4352 bytes. Fig. 2 plots the packet-level throughputs of sources 1 to 4 against time. It can be seen that the throughput of each source exhibits transient behaviour initially and then reaches a steady value, which is close to its fair apportionment.

Conclusion: In this Letter, we have demonstrated that by operating FPD independently and locally in each outgoing link of each node of a network, the interaction between FPD and explicit-rate flow control leads to high and fair packet-level throughput. With max-min fairness maintained, misbehaving sources cannot take advantage of co-operative sources.
Strategies for updating link states in QoS routers

A. Ariza, E. Casilari and F. Sandoval

A crucial aspect of quality of service routing is the frequency with which the nodes have to exchange information about the availability of network resources. The authors classify and compare several strategies for triggering the update process in the nodes. Simulating an actual wide area network as a test-bed, it is shown that the triggering must be based on the relative variations of the available bandwidth in the links, in contrast to using the current available bandwidth or considering an absolute threshold or a policy of periodic updates.

Introduction: Owing to the progressive integration of real time services on internet protocol (IP), traditionally supported on connection oriented networks such as telephonic networks or ISDN, the internet must face the challenge of providing quality of service (QoS) requirements on a connectionless infrastructure. QoS on IP will result from the concurrence of different controls at different levels: packet classification and scheduling, resource reservation, traffic shaping and policing, call admission control and QoS routing. Current IP routing protocols (e.g. OSPF, BGP or RIP) perform the path selection exclusively from topology information, normally minimising the hop count. In contrast with these 'best-effort' protocols, QoS routers must also consider the resource availability as well as the QoS requirements of the flows, to proceed to select a feasible path. This implies that nodes have to exchange information on the present link states. However, if the routing information is exchanged every time the value of the QoS metric changes, it will cause a great burden for the network links and routers, consuming link bandwidth and CPU cycles of routers. In this sense, a tradeoff must be achieved between the need of providing updated link information and the cost of broadcasting and processing update messages.

In this Letter we propose, classify and compare different strategies to trigger the update process in QoS routers. The comparison is performed in a realistic environment in terms of the provisioned QoS and the update rate that the different methods offer.

Strategies for updating link states: In QoS routing schemes, the metric that imposes the cost of a link is normally chosen to be the available bandwidth, as it is strongly related to other QoS parameters [1]. Moreover, using packet scheduling techniques such as weighted fair queuing, delays and loss constraints are guaranteed by means of a bandwidth reservation.

To determine the exact moment in which QoS routers must broadcast the present link state to the rest of the network nodes, we consider three general methods:

(i) Periodical update: according to this simple strategy, which is very common in the literature [1], the actualisation of the available bandwidth is periodically performed regardless of possible changes in the resource availability. The router just has to implement a temporisation of fixed period T.

(ii) Update based on a fixed threshold [2]: a simple way to limit the range of non-updated changes is to establish a fixed threshold for the absolute variations of a state parameter with respect to the last notification. If this threshold is surpassed, the update process is triggered. For available bandwidth, this policy guarantees at any moment that parameters are bounded within the range

\[ B_{ij}' - \Delta B \leq B_{ij} \leq B_{ij}' + \Delta B \]

where \( B_{ij} \) is the present available bandwidth of the link between the nodes \( i \) and \( j \), \( B_{ij}' \) is the last notified value and \( \Delta B \) is the triggering threshold.

(iii) Update based on a relative threshold: in this case, the relative variation of the available resource is considered [2]. Thus, the changes of \( B_{ij} \) are notified when \( B_{ij} \) varies more than a percentage \( T_h \) since the last notified value \( B_{ij}' \). This assures that until a new notification the value of \( B_{ij} \) is limited in the interval

\[ B_{ij}' \cdot (1 - T_h / 100) \leq B_{ij} \leq B_{ij}' \cdot (1 + T_h / 100) \]

A variation of the previous strategies can employ, as the parameter for triggering updates, the utilised bandwidth \( UB_{ij} \) which is obviously defined as

\[ UB_{ij} = C_{ij} - B_{ij} \]

where \( C_{ij} \) is the link capacity between the nodes \( i \) and \( j \).

Simulation and results: For comparison purposes, the commercial network of MCI internet provider was simulated. This network consists of 19 nodes and 77 OC-3 (155Mbit/s) and T3 (45Mbit/s) links. Similar results were obtained for other actual topologies. We considered constant bit rate (CBR) traffic sources the bandwidth of which was randomly chosen between 1 and 5Mbit/s according to a uniform distribution. The call holding time was exponentially distributed with a mean of 1200s. The traffic load was uniformly distributed among the nodes and was kept stable for each simulation (~5 x 10¹⁰ simulated seconds). For each path decision in the routers, it was considered that the cost of each link linearly depends on the available bandwidth, although similar tendencies were observed if other cost functions were utilised [3]. To compare the different strategies for updating, we measured the bandwidth loss probability (Pr), defined as the ratio between the rejected bandwidth and the global bandwidth that the sources demand of the network. As an estimation of the overhead introduced by each strategy, we also estimated the update rate, considered as the global number of actualisation messages per second.

---

**Fig. 1.** Bandwidth loss probability against call rate for different updating strategies

- ■ - \( T_h = 40\% \)
- ▲ - \( T_h = 80\% \)
- Numbers - \( T = 60 \)
- â�� - \( T = 240 \)
- × - \( \Delta BW = 5\text{Mbit/s} \)
- - × - \( T_h = 10\text{Mbit/s} \)