

CH820010101 Peter M Klett/Zurich/IBM
Gero Dittmann

Load Balancer - Switch - Interface

The interface between a network processor load balancer and the switch poses some special problems if the traffic from the balanced network processors should be accumulated into a single switch port:

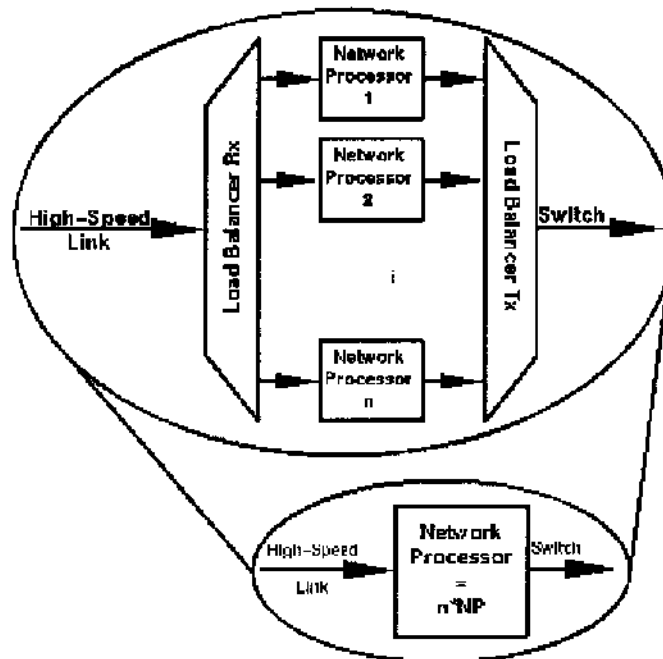


Fig 1

1. From the network processors towards the switch:

A switch supports several traffic priorities and a network processor maintains a virtual output queue per supported switch priority per switch output. The switch gives grants to the network processor about which queue is allowed to send. The switch can also quantify the amount of traffic it allows a network processor to send in the form of credits. The question is, how a load balancer between the switch and a number of network processors forwards this flow control information to the network processors, ensuring that the available bandwidth is fairly shared among them. It cannot just broadcast the information because a grant might allow only one cell to be sent, while broadcasting might lead to at least one cell per network processor to be sent.

2. From the switch towards the network processors:

Packets arriving from the switch are fragmented into cells. If the load balancing is to be done based on upper layer headers, e.g. IP/TCP, these headers are only found in the first cell (if they fit into one cell at all), i.e. subsequent cells cannot be parsed for this balancing information.

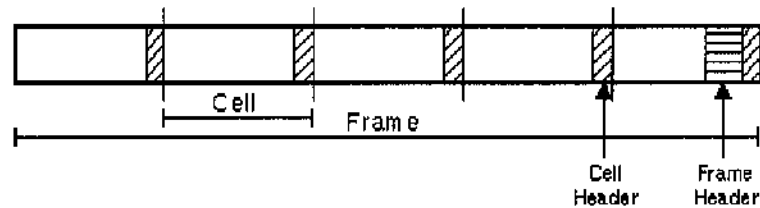


Fig 2

Proposed Solution:

1. The load balancer implements a round-robin scheduler between the network processors per priority per switch output to forward grants/credits from the switch: For each priority-output-combination it stores the last network processor that has sent a cell of this queue. Next time the switch allows traffic for this queue again, the next network processor in round-robin fashion gets a grant signal from the load balancer. With credits, the round-robin forwards only one credit to a network processor at a time and then goes on to the next network processor. It stops when no credits are left for the respective queue. If a network processor does not have anything to send then it is skipped. Other scheduling algorithms can be used as well: e.g. if complete frames should be forwarded from one network processor before advancing to the next network processor then deficit round robin could be used. In this environment, a round-robin scheduler is fair because the cells are all the same size and thus every network processor gets to send the same amount of traffic as long as they have traffic to offer. Furthermore, the load balancer on the link side of the network processor's supplies the network processors with equal traffic loads. The load balancer has to buffer only a few cells for speed reasons and to be able to receive cells from all network processors in parallel. These cells are then scheduled for transmission to the switch in the same manner as the network processor does it internally. Apart from that, no buffers are required in the load balancer but the buffers in the network processors are used.

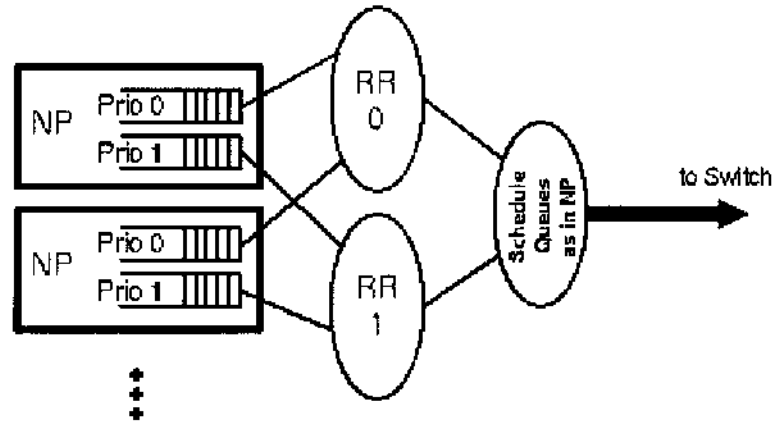


Fig 3

2. The balancing decision is based on a flow ID contained in the first cell of a switch frame. This could be information in the packet header, e.g. the TCP/IP five-tuple (source and destination address, layer 4 protocol number and source/destination ports), or--since this header stack might not fit into the first cell of a frame completely--a flow ID in the switch frame header which is put there by the sending network processor. The balancing decision that has been made for the first cell of a frame (a first cell is identified by flags in the cell header) is stored in the load balancer. For all arriving cells that are not the first in a frame, the decision that has been made for their first cell in frame is looked up and also applied to them. This is called a "Sticky Decision". The lookup index can consist of correlator, source blade, multicast indication, and priority information in the cell header.

This lookup index is only used for correlating cells of a frame while the balancing decision for the first cell of a frame is based on a flow ID. This distinction is necessary because cells from different frames may belong to the same flow.

Mechanisms that change the association of a flow to a network processor, such as Flow Time Outs, Reassignments, and Spraying, are only allowed for a first cell of a frame. For subsequent cells of the same frame these mechanisms must be overridden. It is suggested to identify the flow affiliation of a frame by parsing the LID or MID field (unicast/multicast), respectively, or the frame header extension. The sending network processors should write values there that guarantee a good balancing resolution and flow integrity. All other possible header fields are too small - their use would result in poor balancing results.

Disclosed by International Business Machines Corporation