METHOD AND ARRANGEMENT IN A PACKET SWITCH FOR CONGESTION AVOIDANCE USING A COMMON QUEUE AND SEVERAL SWITCH STATES

A packet switching network has switches and sending/receiving entities interconnected by links (L6) utilized as paths between different users. A switch has an ingress part (11) with a buffer including virtual queues (VQ1, VQ2, VQ3). These are connected to the link (L6) and to a switch core. The queues have threshold detectors (TH1, TH2, TH3) and are connected to states (STA - STD). When the queue (VQ1) is congested by packets the path occupying the individually greatest part of the queue is noted. If noted for the first time the path is stored in a free one of the states (STA,STD) and corresponding sending entity is halted (XOFF). If the same path is noted again on repeated congestion a bandwidth value (3) is counted up. A chronological order for the states (STA-STD) is established. When all states (STA-STD) are occupied an older half of the states is selected, the path with the lowest count value is further selected, its state is purged and the corresponding sending entity is released (XON).
Published:
— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
METHOD AND ARRANGEMENT IN A PACKET SWITCH FOR CONGESTION AVOIDANCE USING A COMMON QUEUE AND SEVERAL SWITCH STATES.

TECHNICAL FIELD OF THE INVENTION

The invention relates to control of flow of data through a switching network. More specifically, the invention relates to data-traffic regulation in a packet switching node for avoiding congestion.

DESCRIPTION OF RELATED ART

One principle for electronically conveying and directing data from a sending entity to a receiving entity through a switching network is known as packet switching. Data is aggregated into packets which carry data and overhead. The overhead comprises e.g. addressing information used by the switching network for conveying the packet from the sending entity to the receiving entity.

A packet switching network has nodes interconnected by transmission links. A node may in turn comprise one or more switch elements interconnected by internal transmission links. There exist different internal structures of switch elements as well as of networks. Such structures are known as e.g. fabrics and topologies, respectively.

A typical node has several switch elements, each having e.g. a crossbar switch fabric. The switch elements are interconnected by transmission links forming an internal network within the node. Packets traversing through the node from an input to an output follow a predetermined route called a path.
For many types of fabrics or topologies, there are scarce resources in a switch element or a network, which have a number of inputs and outputs, called ports. Certain transmission links are shared by several users and may become congested with data. To assure a reasonable data throughput, buffers are arranged in ingress parts at the inputs of the switch elements.

Although efficient, congestion and loss of data may occur due to limitations in the number of viable buffers. One scheme for overcoming such problems is to employ flow control.

Each ingress part of the switch element has a number of the buffers arranged as a number of logical queues, referred to as "virtual output queues". The virtual queue concept solves a problem known as Head-Of-Line Blocking, where packets destined for one congested egress part of the switch fabric are blocking later packets destined for another egress part of the switch fabric.

Each virtual queue has a threshold detector to indicate an emerging congestion condition. At a certain queue threshold level, it is likely that arriving packets will overflow the virtual queue. In order to prevent overflow, a flow control mechanism halts packets at the source, so called flow turn-off.

When a congestion condition ceases, halted packets must be released from the source, so called flow turn-on. Flow turn-on can be accomplished through a timer located at the source. After a certain time interval, it is assumed that the congestion condition has ceased. The timer resets the halt state of the source, and transmission is thus resumed.
This solution however results in inefficient usage of switching resources and poor overall performance of the node.

Another approach for achieving flow turn-on is to monitor congestion in the switch elements, and send a release signal (XON) to sources having halted packets when the congestion condition ceases. Halt states are stored within the switch elements. Each state is associated with a certain path relating to halted packets. The switch element thus remembers paths for which halt signals have been sent. The state is used for creating a release signal corresponding to a previously sent halt signal when the congestion condition ceases. After the release signal is sent, the state is purged and is ready for reuse. The number of paths in a network may be large and it is a difficulty due to physical and/or cost constraints to manage all paths.

**SUMMARY OF THE INVENTION**

The present invention deals with the above problem to make efficient use of available states in a switch, i.e. to purge and reuse states for new paths in an efficient way.

Another problem is to keep the number of states reasonably low in the switch elements.

Still a problem is how to make use of the states with a low hardware complexity.

According to the invention the problem is solved in the following manner. The paths are stored in a queue of the switch. A sending entity for a path occupying the individually greatest part in the switch queue is halted when the queue is congested. The bandwidths of halted paths are successively noted in different states in the switch. A
part of the states, representing the oldest ones, is selected. When all states are occupied and a further state is required, the state among the selected states having the smallest bandwidth is purged and is reused.

5 Somewhat more in detail the problem is solved by storing frames of the different paths in the switch queue. A halt signal is sent to the sending entity when a queue threshold is exceeded. The halted sending entity belongs to the path occupying the individually greatest part of the queue. The bandwidths for the halted paths are successively updated in the different states and the states are in turn noted in chronological order. An older part of the states is selected. When all states are occupied and a further state is required, the state for the path having the smallest bandwidth in the older part of states is purged and reused. This happens when the path of an unhalted sender occupies the greatest part of the queue. The bandwidth of a path is measured as the number of times that path has been found to occupy the greatest part of the switch queue.

20 The invention will now be explained in more detail, with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a block schematic over a network;

Fig. 2 shows a block schematic over a node having a number of switch elements interconnected in a network. The node is a packet switch since it switches packets;

Fig. 3 shows a block schematic over a switch element of a cross-bar type, having virtual output queues in ingress parts of switch ports of the switch elements;

30 Fig. 4 shows block schematic over a switch ingress part with the virtual queues in more detail, a threshold detector and
a flow control device comprising states, each of which has a counter for counting congestion detection;

Fig. 5 a-c show block diagrams over a virtual queue;

Fig. 6 a,b show block schematics over states;

Fig. 7 shows a flow chart over the present method; and

Fig. 8 shows a flow chart over a part of the method.

DETAILED DESCRIPTION OF EMBODIMENTS

A packet switching network has nodes interconnected by transmission links. A node may in turn comprise one or more switch elements interconnected by internal transmission links. There exist different internal structures of switch elements as well as of networks. Such structures are known as e.g. fabrics and topologies, respectively. This definition is used herein.

A simple packet switching network is shown in Fig. 1. It has nodes N1, N2, N3 and N4 which are interconnected by links L1, L2, L3 and L4. The node N1 has switches SW1 and SW2 interconnected by a link L5 and also connected to sending/receiving entities SR1 and SR2 respectively. The switch SW1 is connected to an end user EU1 and the node N1 has further elements as is shown below on Fig. 2. The nodes N2, N3 and N4 also has switches and sending/receiving entities as shown on the figure.

For the sake of simplicity, the invention is described in the context of the node N1 within a packet switching network of Fig. 1. Although it is assumed herein that the invention is applied in a node and that paths have node boundaries, it should not be regarded as limiting the scope of the invention. A node and paths may have a broader definition than those described in the present embodiment without departing from the spirit of the invention.
A typical node has several switch elements, each having e.g. a crossbar switch fabric. A crossbar switch fabric is a matrix of rows and columns that constitute inputs and outputs, respectively. The rows and columns are interconnected as needed by controlling active elements in the cross-points of the matrix of the switch fabric. The switch elements are interconnected by transmission links forming an internal network within the node. Packets traversing through the internal network of the node from an input of the node to an output of the node follow a predetermined route through the internal network. The route is called a path, and all packets following the same path are said to belong to the path. The path starts with the source (or generator), and ends with the destination (or sink).

For many types of fabrics or topologies, there are scarce resources in a switch element or a network. A switch element or a network has a number of inputs and outputs, called ports. Certain transmission links are shared such that data between different inputs and outputs is traversed over one and the same transmission link. These may become congested with data and are thus scarce.

Fig. 2 schematically depicts the node N1 comprising the two interconnected switch elements SW1 and SW2 and also switch elements SW3 and SW4 connected by links L6 and L7. The switch SW1 is further connected to a sending/receiving unit SR3 and the switch SW3 is connected to sending/receiving units SR4 and SR5. The switch elements and the interconnection links constitute a node-internal network. The sending/receiving units are often linked to other nodes or networks, as hinted by links L8 and L9, but may also be connected to an end user terminal such as terminal EU2 connected by a link L10. A route between a source and a destination is called a path such as a path P1 between the entities SR4 and SR3, marked by a dashed line. Another route between the entities SR5 and SR1 is marked as a path P2 with
a dash dotted line. The routes have a common path between the switches SW3 and SW1.

Fig. 3 shows as an example the switch element SW1 in more detail. The switch element has a switch core SC1 connected to an ingress part I1 and an egress part E1. The ingress part is connected to ingress ports IP1 and IP2 and the egress part is connected to egress ports EP1 and EP2.

Data transmitted on the paths P1 and P2, and other paths not shown, to the ingress port IP1 on the switch SW1 can cause congestion in the switch. For alleviating the problem of congestion, and thus assuring a reasonable data throughput through the switch elements and networks, buffers may be arranged in accordance with some principles. Buffers may for example be arranged in the ports at the inputs of the switch elements. Although effective, congestion and possibly loss of data may still occur due to limitations in the number of viable buffers. One scheme for overcoming such problems is to employ flow control.

The ingress part I1 of the switch SW1 has thus a number of buffers. The buffers of each respective ingress part are arranged as a number of logical queues. The buffers of an ingress part is a shared resource. The buffers are dynamically assigned to the different queues on demand. The queues are referred to as "virtual output queues" since each queue is assigned packets relating to a particular output of the switch fabric.

Fig. 4 further shows the ingress part I1 having virtual output queues VQ1, VQ2 and VQ3 connected to the ingress port IP1. Threshold detectors TH1, TH2 and TH3 are arranged for the queues and a selection function SF1 for directing packets from the queues to the switch fabric SC1 is connected to the queues. The threshold detectors are connected to a flow control logic block FC1 comprising a plurality of states STA, STB, STC and STD. Each state has a register with a part 1 for a source address, a part 2 for a destination address and a part 3 for a counter value. The
flow control logic FC1 also includes a control unit C1 reading the queues and controlling the states.

The virtual queue concept solves a problem known as Head-Of-Line Blocking, where packets destined for one congested egress part of the switch fabric are blocking later packets destined for another egress part of the switch fabric.

Each virtual queue has the threshold detector for detecting a predetermined size or utilization of the queue. The threshold detector is set to indicate an emerging congestion condition.

At a certain queue threshold level, it is likely that arriving packets will eventually overflow the queue, the size of which is limited by the availability of buffers belonging to the ingress part.

In order to prevent overflow, a flow control mechanism halts packets at the source, e.g. packets are halted at the sending entity SR4. Paths contributing the most to the queue length of a virtual queue reaching its threshold are identified, and a halt signal (XOFF) is sent to the source relating to that path. In response thereto packets are halted at the source SR4. This is called flow turn-off.

Packets are received at the ingress parts of a switching node. They are placed in their respective virtual queues as determined by their egress part destinations. In the event that the threshold detector of a virtual queue is triggered when a packet is placed in the queue, the path information of each state is checked for a match with the path having the highest representation in the virtual queue, i.e. having the highest byte count.

A match results in the increment-by-one of a counter value stored in the state for which the match occurred. The counter value is an indication of the bandwidth of the related path. It is therefore also called bandwidth indicator henceforth.
Provided there are free states and there is no match, the path information is stored in a free state and a halt signal XOFF is sent to the source. Measures are taken so that a chronological order between states is established with respect to times of seizure of the states, i.e. a point in time when paths were stored in the states. The state may e.g. be linked in a linked list to the other states. The above procedure will be described more in detail in connection with figures 5 and 6.

In figure 5 a, b and c is shown the virtual queue VQ1 with its content of data packets 4 at three consecutive points in time. Packets marked A come from the sending entity SR4, packets B come from sending entity SR5 and packets C to K come from other not shown sending entities.

Figure 6 shows the states STA and STB with a content that is stored after some time of congestion. The sending entity SR4 has the source address SA, the packets A have the destination address DA and the state stores a counter value CA. The entity SR5 has the values SB, DB and CB correspondingly.

Figure 5a shows that the queue VQ1 is congested, the content has reached the threshold TH1 with one of the packets B. A packet C arrives and the threshold TH1 is trigged. In the queue there are 5 of the packets A occupying the individually greatest portion of the queue. The halt signal XOFF is sent to the corresponding source address SA for the sending entity SR4, which stops to send packets. The state STA was earlier free, but now the halted path P1 with the source address SA and the destination address DA is stored in the state STA. Also the counter value is set to CA=1. In a corresponding manner the state STB has been set earlier for the packets B of the path L2 from the sending entity SR5 with a counter value CB=1. In figure 5b is shown that the switch core SC1 has executed the packet D first in the queue. If it is now indicated in the control unit C1 that the queue length goes below the threshold TH1, then a
release signal XON is sent for all the states, states STA and STB, related to the queue VQ1. In the present example we instead presume that the new packet C arrives to the queue VQ1. The threshold TH1 is then trigged once again. The different sets of packets A-K are counted by the control unit C1 and once again the packets A occupy the individually greatest part of the queue. The counter in the state STA is thus set to CA=2. New packets E, A and A first in the queue VQ1 are executed successively by the switch core SC1 as shown in figure 5c. New packets G and F arrive and causes that the threshold TH1 is trigged. The packets A occupy still the individually greatest part of the queue VQ1 for three times and the counter CA is counted up to the value CA=5. Now a packet H arrives and causes that the threshold TH1 is trigged. The packets for the different paths are counted and now it is the path P2 with the four packets B that occupies the individually greatest part in the queue VQ1. The path P2 has already the state STB and its counter is set to CB=2.

In the example above the number of data packets for the different paths A, B ... are counted to determine which path that occupies the greatest part of the queue. This is only for the sake of simple explanation. A better way is to count the total length in the queue that a certain path occupies e.g. counted as the total number of bytes. The age of a state can be noted in different ways, e.g. by a time stamp or by arranging the states in a time loop. It should be noted that in a real situation the flow control block FC1 normally has many more states than shown in figure 4 and that the virtual queues VQ1, VQ2 and VQ3 have many more data packets than shown.

The different states STA, STB, ... are taken into use successively and are noted in the time order they were started by the control unit C1. The number of states is limited and after some time they are all occupied. However, if there are no free states, a state must be purged, i.e. be
made ready for reuse. The manner in which a state is chosen for purging will now be further explained.

The states are divided into two parts, an older part and a newer part, according to the embodiment an older half and a newer half. The older-half of the states, including the staes STA and STB, are evaluated with respect to their bandwidth indicators, which in the embodiment is the counter values CA, CB ... . The state having the lowest counter value will be purged. Should two or more states have one and the same lowest counter value, the oldest gets purged. When a state gets purged, a release signal XON is sent to the source related to the state. In the example above the signal XON is sent to the entity SR5 sending the packets B.

After the state has been purged, it is ready for reuse, and the path of the packet placed in the virtual queue is stored in the purged state in the same manner as described above.

Figure 7 shows a flow chart over the method that is described above in connection with the figures 4, 5 and 6. The method starts with that the ingress part I1 receives a data packet, a step 701. In a step 702 is investigated whether the virtual queue VQ1 has reached the threshold TH1. If not so, an alternative NO1, no action is taken according to a step 703. A new packet is received according to step 701. If the queue is congested, an alternative YES1, the different portions in bytes that the different paths A, B, C, ... occupy of the virtual queue VQ1 is counted, step 704. In a step 705 is selected the one of the paths that occupies the individually greatest part of the virtual queue VQ1. In a next step 706 is investigated whether the selected path is already stored in one of the states STA, STB, ... . If that is the case, an alternative YES2, the counter value for the path is counted up one step and a new data packet can be received, step 707. In an opposite alternative NO2 it is investigated in a step 708 if any of the states STA, STB, ... is free. If so, in an alternative YES3, the selected path is stored in the free state, step 709. According to a step
710 the halt signal XOFF is now sent to the source address
for the selected path. This path is then registerd in time
order with the other halted paths in a step 711 and new data
packets can be received. If in the step 708 no state was
found to be free, an alternative NO3, one of the earlier
occupied states must be purged and be reused. This is
performed such that in a step 712 the states are divided
into two parts, an old part and a new part. In the
embodiment the states are more specifically divided into an
old half and a new half. In a step 713 the old part of the
states is examined and the state or states with the smallest
bandwidth are selected. The bandwidth is in the embodiment
measured as the counter value CA, CB, ... . In a step 714 is
examined whether two paths have the same bandwidth. If this
is the case, an alternative YES4, the oldest one of the
states is purged in a step 715. In a step 716 the release
signal XON is sent to the source for the path in the purged
state. This state is then reused in accordance with the
steps 709, 710 och 711. If in the step 714 there is only the
selected state with the lowest bandwidth according to an
alternative NO4, this selected state is purged in a step 717
and the release signal XON is sent in a step 718. As above
the purged state is then reused in accordance with the steps
709, 710 och 711.

25 For the sake of completeness it will be described in
connection with figure 8 what happens when the data packets
in the virtual queue VQ1 are executed in the switch core
SC1. In a step 801 a data packet is sent to the switch core.
It is then investigated in a step 802 whether the virtual
queue VQ1 is still congested. In an alternative YES5 no
action is taken according to a step 803. In an alternative
NO5 all states related to the the virtual queue VQ1 are
purged according to a step 804. In a step 805 the release
signal XON is sent to all the sources sending data packets
via the virtual queue VQ1.
In pseudo-code, an algorithm for carrying out the method would be as follows:

Each time a packet is received at an ingress part:

5 {
  IF (virtual queue > threshold) THEN
    IF (path of packets having highest byte count in virtual queue is already stored in state) THEN
      increase bandwidth indicator in the state
  ELSEIF (all states are used) THEN
    select the older half of the states
    IF (more than one state has the same value in the bandwidth indicator of the selected old states) THEN
      send a release signal (XON) to the source with the oldest state and purge the state
      send a halt signal (XOFF) to the source of packets having highest byte count in virtual queue
  15 }

Each time a packet is sent from the ingress part into the switch fabric:

  IF (virtual output queue < threshold) then
    send a release signal for all states related to the virtual queue
CLAIMS

1. Method in switching a data flow of information packets intended for paths between a respective sending and receiving entity, the method including:

- buffering the packets from the paths in a queue;
- halting a sending entity on congestion of the queue;
- storing the halt condition in a switch state,

characterized in:

- noting the individual portions that different of the paths occupy in the queue;
- halting the sending entity for the path occupying the individually greatest portion of the queue;
- storing the halted path in a free one of the switch states including storing its bandwidth;
- successively updating the respective bandwidth of halted paths as the queue is repeatedly congested;
- establishing a chronological order of the states;
- determining an older part of the states; and
- purging the state for a path having the smallest bandwidth in said older part of the states.

2. Method according to claim 1, characterized in reusing the purged state.

3. Method according to claim 1 or 2, characterized in determining the queue congestion by a threshold.
4. Method according to claim 1, 2 or 3, characterized in that the noting of the individual portions that different of the paths occupy is performed as a byte count.

5. Method according to any of the claims 1-4, characterized in that the state includes a counter field and that the path bandwidth is noted in said counter field as the number of times the respective path has been found to occupy the individually greatest portion of the queue.

6. An arrangement for switching a data flow of information packets intended for paths between a respective sending and receiving entity, the arrangement including:

- a queue device for buffering the packets from the paths;

- a device for halting a sending entity on congestion of the queue;

- switch states for storing the halt condition, characterized in:

  - means for noting the individual portions that different of the paths occupy in the queue;

  - the device for halting has means for halting the sending entity for the path occupying the individually greatest portion of the queue;

  - the switch states have means for storing a bandwidth indicator for the halted path;

  - means for successively updating the respective bandwidth indicator of halted paths as the queue is repeatedly congested;

  - means for establishing a chronological order of the states;

  - means for determining an older part of the states; and
means for purging the state for a path having the smallest bandwidth in said older part of the states.

7. An arrangement according to claim 6, characterized in that the purged state is arranged to be reused.

8. An arrangement according to claim 6 or 7, characterized in that it includes a threshold detector for determining the congestion of the queue.

9. An arrangement according to claim 6, 7 or 8, characterized in that the means for noting of the individual portions that different of the paths occupy is arranged to count the number of bytes.

10. An arrangement according to any of the claims 6-9, characterized in that the state includes a counter field and that the arrangement has means for noting the path bandwidth in said counter field as the number of times the respective path has been found to occupy the individually greatest portion of the queue.
FIG. 1
FIG. 3
FIG. 4
FIG. 7

SUBSTITUTE SHEET (RULE 26)
FIG. 8
**INTERNATIONAL SEARCH REPORT**

**A. CLASSIFICATION OF SUBJECT MATTER**

**IPCC:** H04L 12/56, H04Q 11/04

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

**IPCC:** H04L, H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

**EPO-INTERNAL, WPI DATA, PAJ**

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>WO 0167672 A2 (SUN MICROSYSTEMS, INC), 13 Sept 2001 (13.09.2001), page 4, line 26 - page 7, line 7; page 9, line 14 - page 11, line 21, figures 7,8, claims 1-11</td>
<td>1,3,4,6,8,9</td>
</tr>
<tr>
<td>P,X</td>
<td>US 6515963 B1 (BROTHSHEM, A.V. ET AL), 4 February 2003 (04.02.2003), column 3, line 55 - column 4, line 17; column 13, line 11 - column 14, line 25, figure 3, abstract</td>
<td>1,3,4,6,8,9</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

**Date of the actual completion of the international search**

26 January 2004

**Date of mailing of the international search report**

27 January 2004

Name and mailing address of the ISA/
Swedish Patent Office
Box 5055, S-102 42 STOCKHOLM
Facsimile No. +46 8 666 02 86

Authorized officer
Marianne Engdahl /LR
Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (January 2004)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Number</td>
<td>Date</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>WO</td>
<td>0167672</td>
<td>13/09/2001</td>
</tr>
<tr>
<td>US</td>
<td>6515963</td>
<td>04/02/2003</td>
</tr>
<tr>
<td>US</td>
<td>6359861</td>
<td>19/03/2002</td>
</tr>
</tbody>
</table>