Abstract: - Using optics to build large scale packet switches has been proposed in the literature due to potential advantages regarding power consumption and scalability. The focus of this article is optical packet switches that operate in a time slotted synchronous mode. Since the timeslots are fixed length the incoming traffic needs to be aggregated into optical formatted packets (aggregation process). Optical formatted packets are announced to a central scheduler that controls the permutation of the optical switch backplane. In this article the aggregation process is investigated in conjunction with the central scheduling algorithm in order to make a quantitative evaluation.

Key-Words: - Modeling, simulation, optical packet switch, scheduler and aggregation process.

1 Introduction

Traditionally large packet switches has been build using electronics in both the line cards and the switch fabric. Packets would be segmented into small size segments or cells (e.g. 64 bytes) and send on high speed serial links to the central switch fabric. After passing through the switch fabric the segment would be reassembled in an output buffer at the outgoing line card. As transmission line rates have increased so has the power consumption of the high speed serial links. This has made the switch vendors to split the high speed serial lines into a number of parallel lines with lower bandwidth. The growth of the internet traffic and the use of DWDM imply that the number of line cards also has to be enlarged. Power consumption restrictions and the physical size of the line cards also imply that the packet switch should be spread over a number of physically distinct racks. E.g. one rack holds the switch fabric while other racks hold the line cards. As the number of line cards increase it has been considered and option to use optical transmission between the line cards and the switch fabric in order to reduce the number of connections and potentially the power consumption. This approach reduces the problems of the many parallel interconnects but it also add complexity due to the large number of optical to electrical conversions (and vice versa). Using an optical packet switch as the central switch fabric reduces the number of OE and EO internal conversions by a factor of two compared to the switching system using only optics for transmission on the backplane.

Using an optical packet switch as the central switching crossbar implies some changes on the line cards in contrast to ordinary packet switch systems. Since the optical switch matrix requires some overhead time to change configuration using long timeslots (optical packet length) compared to traditional systems is necessary. The optical packet length should not be made too long since poor efficiency might be the results due to low filling ratio of the optical slot. Due to the long optical packet length, making it possible to contain multiple IP datagram’s in one optical packet is considered a must in order to avoid inefficient transport across the switch fabric. Supporting multiple IP datagram’s in one optical packet requires division of the optical packet into segments. This approach is illustrated on Figure 1.

![Figure 1 Optical packet format](image)

The optical packet contains two segments from IP datagram A and one from IP datagram B and C. All segments are destined for line card 1. The total number of segments is fixed at k. Each segment contains a small overhead which is used by the receiving line card to assemble and the original IP datagram. The overhead is explained on Table 1.

<table>
<thead>
<tr>
<th>Overhead name</th>
<th>Length (bits)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of segment</td>
<td>1</td>
<td>Indicates if the segment carries the first part of an IP datagram</td>
</tr>
<tr>
<td>End of segment</td>
<td>1</td>
<td>Indicates if the segment carries the last part of an IP datagram</td>
</tr>
<tr>
<td>End pointer</td>
<td>$\log_2$(Optical packet length in bytes)</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

If the carried segment is the last part of an IP datagram a part of the segment might be empty. The end pointer indicates the last valid byte in the segment.

2 Article overview

The rest of this article is organized as follows: In section 3 the architecture under evaluation is described together with the optical formatted packet used. Section 4 presents the modeling work in details. Section 5 contains the simulated data sets and results. These are followed by conclusions on the work presented in section 6.

3 Architecture description

The system under investigation in this article is depicted in figure 1.

Figure 2 Architecture overview

The (simplified) line cards segments the original packets into fixed length segments and forwards them to the VOQ's were they are aggregated to form fixed length optical packets. When a segment enters an empty queue a timer with value $\tau$ is started. Two different events can trigger the closing of an optical packet. Either it is closed because it is full with segments or either it is closed due to the timer running out. The later case gives rise to a higher optical packet load and therefore a fabric speedup might be needed to compensate for this inefficiency. The queue manager's sends information regarding the queue status to the central scheduler in each timeslot and the central scheduler sends transmission acknowledges to the queue manager in each time slot. Optical packets received at the output line card are stripped from any overhead and the segments are used to reassemble the original data packet. Once a packet is reassembled it will be sent on the outgoing transmission line. The scheduler used in this evaluation is a modification of the original iSLIP scheduler [1]. The modified scheduler i-\Delta SLIP was proposed in [2] to enhance the throughput of iSLIP in switch fabrics with large round trip latency. The aggregation of segments in the VOQ of a line card is further illustrated in figure 2.

Figure 3 Aggregation of segments into optical packets

Three segments arrive with the same output line card destination. The segments are stored in the optical formatted packet and eventually sent through the switch fabric. N output queues are needed at the output line card since all inputs might have un-assembled segments left. However there is no need to have a scheduler on the output line cards since the packets are re-assembled at distinct different times.

The fabric and line cards may run at different clock speeds as illustrated on Figure 2. This speedup factor can be used to compensate for inefficiencies in scheduling or low optical packet load.

Earlier studies [3], [4] shows that the selection of the timeout parameter $\tau$ has a high impact on the average delay that packets encounter in the aggregation stage.

4 Model description

The architecture under evaluation if modelled using OPNET modeller 10.0. The architecture is modelled in a single node where a number of processes are used to instantiate a number of dynamical processes depending on the number of line cards to be evaluated. The overall node design is depicted on Figure 4.
The source is a process model which can be exchanged according to the studied setup. The source typically instantiates a number of dynamic processes which generates the desired traffic profile for each individual linecard. Shown on Figure 5 is a dynamic process which generates packets trains and pauses with geometrically distributed train and packets lengths.

\[
\lambda_{i,j} = \begin{cases} 
\lambda \left( \omega + \frac{1-\omega}{N} \right) & \text{if } i = j \\
\lambda \left( \frac{1-\omega}{N} \right) & \text{otherwise}
\end{cases}
\]

\( \lambda_{i,j} \) is the traffic intensity from input \( i \) to output \( j \), \( N \) being the number of line cards. \( \omega \) is the degree of non-uniformity. The following conditions apply:

\[
0 \leq i, j < N
\]

\[
0 \leq \omega \leq 1
\]

The offered load per input and output port is admissible when \( 0 \leq \lambda \leq 1 \):

\[
\lambda_i = \sum_{j=0}^{N-1} \lambda_i, \quad \lambda_j = \sum_{i=0}^{N-1} \lambda_{i,j} = \lambda \left( \omega + N \frac{1-\omega}{N} \right) = \lambda
\]

The pause lengths and burst length are determined by geometrical distributions.

The segmentation process is illustrated on Figure 6.

When a packet arrives at the segmentation (main) process the FORWARD state is reached. The FORWARD STATE determines which line card the packet belongs to and invokes the corresponding segmentation dynamic process for further processing of the packet. The segmentation dynamic process is shown on Figure 7.
segmentation process it is stored in the internal VOQ queue structure and the scheduler is informed by sending a specific info packet. The segments are forwarded to a dynamic aggregation process depending on the source of the segment (line card). Packets arriving from the scheduler control the emission of optical packets from the queue structure.

The aggregation and queuing process is depicted on Figure 9. Segments arriving are stored in the VOQ based on the line card source and the line card destination. When a segment arrives to a specific queue it is investigated if the segment is the first in an optical packet. If this is the case a self-interrupt is scheduled at time \( \tau \) later. The optical packet can either be closed by receiving enough segments to fill the optical packet or by receiving the self interrupt indicating that a number of empty segments must be carried in the optical packet. This is done by generating and storing a number of dummy segments. The dummy segment generation are informed to the scheduler by sending specific formatted packet.

The scheduler process is illustrated on Figure 10. The scheduler is contained in one main process which receives information from the aggregation and queuing processes. Information sent from the aggregation and queuing processes are possibly delayed in order to simulate latency between line cards and the central scheduler. The scheduler is activated (self interrupt) once per optical time slot. The calculated switch permutation is sent back to the aggregation and queuing process using a special formatted packet.

The scheduler algorithm under study in this paper is presented in [McKeown].

The optical packets are forwarded over the crossbar process and delivered at the sink module. The streams connecting to the crossbar process can be associated with a delay in order to simulate the roundtrip delay between line cards and switch stage.

The sink measures the delay of the individual segments and writes the statistics to an output scalar file for post-simulation analysis.

### 4.1 Measurement and simulation method

Delay is defined as the time from which a packet is completely captured in the input line card until it goes completely out of the output line card. The delay is measured in number of optical slot length (1us). Throughput is measured in each timeslot of the optical switch as the ratio between busy to total output switch ports. Measurements are done when the system has reached steady-state (when possible).

### 5. Simulation studies
In order to verify the model a number of runs have been made where the segment size and optical packet size are equal. This makes it possible to compare with the results obtained in [McKeown] and [Minkenberger] for model verification.

Figure 11 32x32 5-SLIP average burst size 1
Figure 11 and Figure 12 shows the throughput for two setups were the optical packets contain exactly one segment. It is observed that the system is rather sensitive to increasing average burst lengths even for the same average load. This conforms well to the existing results on iSLIP scheduling [1],[2].

Table 2 Simulation setup parameters

<table>
<thead>
<tr>
<th>Simulation setup</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical Packet length</td>
<td>1024B</td>
</tr>
<tr>
<td>Segments per optical packet</td>
<td>16</td>
</tr>
<tr>
<td>Time slot length</td>
<td>1us</td>
</tr>
<tr>
<td>Timeout parameter $\tau$</td>
<td>45us-305us</td>
</tr>
<tr>
<td>Offered load</td>
<td>75%</td>
</tr>
<tr>
<td>Arrival process</td>
<td>On-Off model</td>
</tr>
<tr>
<td>On and Off states</td>
<td>geometric distributed</td>
</tr>
<tr>
<td>On state average length</td>
<td>16 segments (1 optical packet length)</td>
</tr>
</tbody>
</table>

The result of the simulation can be inspected on Figure 13

Figure 12 32x32 5-SLIP average burst size 30

In order to investigate a more realistic setup the parameters shown in Table 2 has been used:

Figure 13 32x32 5-SLIP under various $\tau$ values

It is observed that selecting $\tau$ to a value of approximately 100us gives a minimum average delay value. The reason for this behaviour should probably be found in two different mechanisms. For values of $\tau$ lower than the optimum, optical packets are filled incomplete and thus wasting bandwidth across the switch fabric. For values of $\tau$ larger than the optimum, optical packets are closed due to full filling and thus the first segment in an optical packet has to wait for the last segment arriving. This behaviour is further illustrated on Figure 14.
Figure 14 Average slot load under various $\tau$

Selecting a fixed $\tau$ at the optimum delay is not a valid approach. Doing this will limit the admissible traffic seriously with bad performance as result. Using a variable $\tau$ based on the overall system state is considered as an option for optimum performance.

6 Conclusion

This article presents an optical packet switch system with aggregation of optical segments into optical packets. The model is presented in detail and simulated verification data is documented for some simple cases. Furthermore the importance of the aggregation timeout parameter $\tau$ has been investigated. Is has been illustrated that selecting a fixed $\tau$ can result in poor system performance. Investigating the system performance for other traffic profiles is required to establish a useable mechanism for choosing $\tau$.

References:


