1 Introduction

PCI Express 3 was announced in August, 2007. Like its predecessor, it sought to further the insatiable demand for performance by doubling throughput. Unlike its predecessor, it did not do so by simply doubling the bit rate.

It is well understood that performance of a communications connection is a function of both throughput and latency. Beyond cut-through communications protocols such as Infiniband, very little work has been done to address latency - especially not in peripheral interconnects within a computer. In fact, an hypothesis of my research is that if anything, latency has grown worse as a largely unavoidable consequence of certain design decisions required to boost throughput.

Like processors in the past, throughput was conventionally advanced by building faster and wider buses - but like processors, there is a definite limit to the success of this strategy.

The physical dimensions of a bus limit frequency both because of dielectric loss and because power consumption is a function of both frequency and capacitance. PCI Express can perhaps double frequency once more before costly materials such as Teflon become necessary in manufacture.

Optical technologies show great promise as solutions for improving both throughput and latency but have evident drawbacks, notably the lack of an equivalent of RAM.

2 Related work

The importance of latency in communications performance is generally well understood. [MVCA97] showed how host overhead influenced performance in Myricom hardware, and described a microbenchmarking technique.

The HPC community makes heavy use of low-latency protocols with cut-through routing such as Myrinet and Infiniband. Liu et al. showed in [LCW+03, LWP04] that in such systems, latency is in the order of 5 µs and that RDMA polling times can be as low as 600 ns, figures low enough that interconnect latency could be a nontrivial factor.

Data vortex [LSLL+03] uses deflection routing (concentric rings of fibre with periodic entries and exits) to avoid the need for buffering packets. [SSLLB05] describes a fully populated 12 × 12 data vortex. It’s a novel approach, but difficult to build because it doesn’t support variable length packets, every fibre must be exactly the same length, and latency through the fabric is potentially unbounded.

A PCI Express interface for the data vortex was described in [LLW07], but the paper assumes fixed length packets (as required by the data vortex), and the use of only one photo detector per agent limits sustained throughput to that of one wavelength.

SPINet [SLB05] is an optically addressed, self routing architecture intended for use in highly integrated situations. It avoids delay lines and deflection routing, but resolves contention by dropping traffic.

Micro-ring resonators can be used to build very efficient, low power switching elements [KKL07]. There is a lot of work relevant work that builds upon micro-
ring resonators refs. Micro-ring resonators are exciting, but are difficult to fabricate reliably, are very sensitive to temperature and by their nature are not well suited to WDM.

[KL06] and [KKL06] describe a low latency shared memory suited to large multiprocessor environments. Corona [VSM+08] is an on-chip interconnect suitable for chip multiprocessors.

[GDM+05] describes a local area WDM optical interconnect which uses an hybrid approach of using an electronic control plane with an optical data plane, similar to what is proposed in this thesis.

[Tuc06a] and [Tuc06b] consider optically switch IP routers with reference to buffering requirements, power consumption and physical density. Tucker concludes that due to the lack of optical RAM, electronics will remain superior to all-optical implementations for the next decade or two, although McKeown made an argument in [McK04] for IP routers with no buffering.

Recent work describes an optical [HBY+08] memory interface, and IBM’s TeraBus [KDK+05, SKD+06] describes a complete chip-to-chip packaging transmission system.

3 Factors of performance

3.1 Throughput

Bus throughput can be augmented either by increasing width or clock frequency up to a limit imposed by a combination of clock skew, package pin count and signal integrity.

PCI Express avoided these problems by employing high speed serial lines in which a clock is encoded in the data. Multiple lanes can be ganged together to provide required throughput, and de-skewing can be done on a lane-by-lane basis.

As with conventional PCI, PCI Express doubled throughput once by doubling frequency to 4 Gb/s at a signalling rate of 5 GT/s, \(^1\) [PS06] which is low enough for signal integrity to be acceptable in FR4 printed circuit boards provided care when designing transmission lines.

Doubling the signalling rate again brings the signalling rate to the point where signal integrity becomes challenging in FR4 PCBs. To manage this, the proposed new revision of the PCI Express standard has changed the coding method from 8b10b (a coding method common in high speed serial communications protocols) to 128b130b, reducing overhead from 25% down to about 2%. For almost double the throughput, the signalling rate increases by only 60% to 8 GT/s.

Like processor clock frequency in the past then, it seems likely that easy performance increases are things of the past.

3.2 Latency

Latency is an important factor in performance [Che96] and, in general, bandwidth has grown at a faster rate than latency has been improved. This was at least been established for disc, CPU, memory and networks in [Pat04], but Patterson’s paper was silent about local interconnects.

3.2.1 Sources of latency

The PCI protocol [PS02] was introduced in 1998 for use in personal computers to improve throughput and solve certain shortcomings of earlier bus protocols. The original protocol was specified to 33 MHz, and provided up to 16 clock cycles for slow peripherals to respond to a transaction.

PCI-X was introduced in 1998 to further enhance throughput [PS00]. It was rated to a somewhat higher maximum clock speed (133 MHz), and provided an option for split transactions. Split transactions allow a target to shoulder responsibility for completing a request by initiating a new transaction containing the requested result at its own convenience. Split transactions were necessary to accommodate devices that couldn’t respond within the 16 cycle wait state limitation at 133 MHz, but had the unfortunate side effect of increasing overhead (two bus transactions instead of one) and removing the upper bound on completion latency.

\(^1\)The unit gigatransfers-per-second accounts for the overhead imposed by coding method.
PCI Express made a radical departure from conventionally clocked parallel buses by introducing ganged, high speed serial links. The standard borrowed heavily from the Physical Coding Sublayer of Gigabit ethernet, using the same 8b10b coding. Only the bit rate was increased from 1.25 GT/s to 2.50 GT/s. Transactions became messages encapsulated in packets, and split transactions perforce became compulsory where they were optional under PCI-X.

Conventional PCI and PCI-X used parity to detect bus errors. Data integrity is checked as data arrive, so an agent (bus device) can begin to process a transaction before the transaction is ended. PCI Express is packet oriented and as with Ethernet, the entire transaction is protected with a CRC. A PCI Express agent must buffer the packet until the entire transaction has arrived before it can begin to process the transaction because it is impossible to tell where the error is, if one is present.

Even if it were unavoidable, PCI Express probably did a lot to hurt latency.

3.2.2 Significance
Conventionally, bus latency was probably insignificant relative to the speed at which peripherals operated. As peripherals become faster, the significance of bus throughput and latency increases.

[MVCA97] showed that system overhead was the most significant factor in performance of Myrinet adapters. [LCW+03, LW04] showed that RDMA operations over Infiniband and Myrinet are in the order of microseconds, with polling operations as little as 600 ns.

Algorithmic share trading accounted for an estimated third of all trades in 2006 [Gro06], up to over 50% projected into 2010. Latency, even into the microseconds translates into a competitive advantage [Thi08].

In his 2008 Hot Interconnects keynote, Andrew Bach said that the NYSE data centre already uses multiple 10 Gb/sec networks, and would readily use multiple 100 Gb/s networks if they were commercially available because trade latency is so important. By comparison, a sixteen lane PCI Express v3 link offers only about 114 Gb/s for payload, only marginally more than the network’s link speed. The host interface of 100 Gb/s host adapter would have to be very efficient before it could saturate its link.

3.3 Power consumption
The high speed serial transceivers used by PCI Express consume more power than the equivalent parallel bus because they contain logic not required on parallel buses (such as clock recovery, CRC logic and buffer management). By way of example, the power consumption of each transceiver in the Virtex II is 300mW [Xil04].

Consumption is further increased by the transition density required to maintain clock synchronisation and DC balance, and by the greater amount of logic required to implement a PCI Express end-point.

Power consumption is also a function of frequency. Faster electronic interconnects will not help contain or reduce power consumption in data centres.

4 Optical solutions
Given issues of throughput, latency and power consumption taken together, it seems likely that existing electronic interconnect technologies are not capable of meeting the demands of emerging and near-future peripherals, and drastically limit the effectiveness of peripherals such as accelerators. A solution may lie in an optical interconnect.

4.1 Properties of optical technologies
Photonic technologies have a number of features which make them attractive as a possible alternative to electronic transmission lines.

The most obvious benefit is sheer bandwidth. Unlike loss in electronic transmission lines, loss in optics is a function of (large) distances only and not of modulation frequency. Latency due to buffering is absent because there is no equivalent of optical

---

2Based on 64 bit addressing and 128 byte TLPs. Although a TLP can theoretically carry a 4 kB payload, 128 bytes is standard because few chipsets have buffering for larger.
RAM. Where electronics use power during switching, lasers (including SOAs, described below) use power to maintain the inversion layer of the lasing medium, so power consumption is independent of modulation frequency.

Optics provides not just a potential escape from all of the problems outlined above, but other benefits. Where electronic switches require at least one switching element per bit or lane, wavelength division multiplexing can be used to combine multiple lanes so that a single optical switching element can switch all lanes at once. Further, optical switching elements are bidirectional - light can go in both directions at once.

Light doesn’t radiate electromagnetic interference, at least, not of the kind that matters to the FCC. Transmission lines carrying high frequency signals make electromagnetic compliance harder. Optical interconnects promise escape from a problem which has conventionally been very difficult to manage. [ban07]

4.2 Optical building blocks

The field of photonics has existed for a long time and offers a wide array of components to choose from. This section describes the components that may be useful for the purposes of this research.

Laser transceivers are a commodity item available in compact pluggable modules. For moderate speeds, laser light can be modulated directly by end-point electronics. Above 10 Gb/s, it is more common to modulate a constant wave light source with an external component such as a Mach-Zehnder interferometer.

Light can be switched using a wide variety of devices, including Semiconductor Optical Amplifiers (SOAs), Micro Electro Mechanical Systems (MEMS), Lithium Niobate Mach-Zehnder Interferometers and Spacial Light Modulators. Each have their strengths and weaknesses.

An SOA is a laser without mirrors. It generates very little light on its own, and outputs light from an external source as a function of the bias current applied to it. SOA switching time is related to carrier life-time, which is in the order of 1 ns. Put together, SOAs make excellent switching elements, and have been used successfully in a number of applications [AWG+07]. We will use these in the proposed thesis.

As described in that paper, an optical switch fabric can be constructed using pairs of SOAs functioning as complementary on-off switches, connected using optical combiners and splitters. The intrinsic gain capability of using SOAs as switches is useful for compensating for passive loss in the combiners and splitters required for coupling the switch fabric.

4.3 Optical architectures

Interconnect networks have been well described and are the subject of text books like [DT04] and [DYN03]. Many of them, notably Beneš [Ben62] and Clos [Clo53] networks lend themselves well to implementation in optics. Eng Tin Aw et al. described a practical implementation of such a medium degree (32×32), fully interconnected, non-blocking SOA based switch [AMW+08].

Fully interconnected switches are possible, but expensive and large due to the number of SOAs involved (640, in the case of the aforementioned 32×32 switch.) Although, in principle, any device can communicate with any other device in personal computers and servers, this very seldom happens because it is unusual for any two peripherals to speak a mutually compatible protocol.

Therefore, an N×1 switch is likely sufficient for the vast majority of cases, and inter-device communication can be implemented in two passes through the switch fabric for the few cases in which direct, peripheral-peripheral communication is necessary.

Scheduling a device’s access to an optical switch fabric is made considerably easier by dividing time into fixed length slots at the potential expense of increased latency and fabric utilisation. An important part of the work in this thesis will be to find a suitable choice for the time slot size, and assess the performance trade-offs involved.

5 Thesis and desired outcomes

The preceding sections introduced and provided background on the problem which my research hopes
to address and some of the tools that may be used to do so.

This section sets out the claims that my research seeks to prove, and describes the contribution that it will make.

The hypothesis of my research that an unbuffered optical interconnect can:

- be viable
- perform at least as well as conventional electronic interconnects
- outperform electronic interconnect bandwidth
- reduce latency and therefore increase performance

by virtue of the loss profile inherent to photonic components.

For the sake of simplicity, the research will begin by considering these claims in respect of the peripheral interconnects found in standard COTS (commercial, off-the-shelf) PC server hardware. PCs are self contained (they lend themselves well to the proposed architecture), readily available (ease of experimentation) and collectively represent a highly significant proportion of the world’s aggregate computing power (economically worthwhile).

Front-side bus (CPU, memory, etc) interconnects are out of the scope of this research, but are logical and worthwhile targets for future work. Similarly, high performance computing may benefit (especially clusters) from the techniques covered in this thesis, but are outside the scope of the research.

The research is not concerned with photonic components, and will use the existing technology described in section 4.2 as discrete components. The integration required to make the proposed architecture commercially viable, such as building multiple SOAs together with suitable semiconductor waveguides is left to others.

There are significant implications for the way future computers and applications are built should the thesis be proved.

The experimental work aims to support a thesis statement similar to the following:

- Unbuffered optical backplanes work and perform better than electronics, or
- Unbuffered optical backplanes work, are comparable with electronics, and could be made to exceed by given development, or
- Unbuffered optical backplanes can be made to work, but are as yet impractical for industrial applications because of given factors, or
- Unbuffered optical backplanes won’t work because of given factors, and an alternative to wider, faster interconnects must be found.

Additionally, some estimation of the point at which power profiles favour optical interconnects over electronic interconnects and/or some estimation of how far the proposed system might scale before a buffered OEO bridge would be required may be made.

6 Methodology

The work begins with several hypotheses, which the experimental work described here is designed to either support, modify, or disprove:

- Peripheral interconnect latency is significant enough to affect application performance
- Buffering together with store-and-forward packet-oriented interconnects are make this latency worse
- The characteristics of photonic technology allow for an unbuffered optical switching fabric without the store-and-forward architecture of existing electronic interconnects, as well as increasing system throughput past that which electronic interconnects are capable
- The interconnect can be designed so that it is otherwise competitive with electronic interconnects

For the sake of keeping things simple and containing expenses, all of these experiments will be carried out on a 1 lane PCI Express link, with the
consequence that bandwidth figures will look low. Real implementations would run with much higher throughput (both higher symbol rates and widths (viz. WDM)).

It should also be noted that in the various demonstrator experiments, the overall system performance can never be greater than the host uplink, and that at best, the performance of devices plugged into the demonstrator can only equal the performance of the same devices plugged directly into the host. The measure of how good the demonstrator is is how close the demonstrator is to native performance.

A close result demonstrates that the proposed interconnect is viable and efficient, and the expectation is that it will carry that performance when the throughput (bit rate and number of lanes) is increased past that of an electronic interconnect.

The rest of this section will describe in more detail the flowchart in figure 1.

6.1 Experiment 1: PCI Express latency

The first experiment is designed to characterise transaction timing on a variety of PCI buses, with a view to establishing the role of interconnect latency and buffering in peripheral performance. The data gathered will be used to answer the following questions:

- Do split completions and/or CRC protected, serialised packet-like transactions make latency worse than conventional parallel immediate (non-split) transactions?
- How much does transaction buffering contribute to the overall transaction latency?
- How much latency is there in the interconnect, and is it significant relative to the speed at which different sorts of workloads (network, disc, video, memory, hardware accelerators etc) operate?

Slow devices like discs and USB devices are not going to benefit from a low latency interconnect. High performance interconnects like InfiniBand and Myrinet might - especially when those protocols employ cut-through switching in order to reduce latency,
yet the interconnects which feed the interface cards are store-and-forward.

As network parameters approach, or even exceed the parameters of a host’s internal interconnect, it seems very likely that the high bandwidth and low latency possible in optical interconnects could be very beneficial. Finally, accelerators and co-processors are peripherals which absolutely depend on low latency in order to be worthwhile. As a class, they have largely been under exploited because the interconnects between peripheral and host processor(s) make accelerators ineffective.

For reasons described in the (separate) proposal for experiment 1, a direct measurement of a PCI bus is difficult. Instead, the latency associated with each segment of the local interconnect can be inferred from a set of differential measurements (microbenchmarks) from an origin to each hop along the path to a target (similar to how traceroute works). The latency due to buffering can be calculated from the link speed and the size of the probe packet.

The data will provide a base-line against which to compare the data from later experiments, and inform certain design decisions about the new optical interconnect. For example, if it turns out that CRC error detection increases latency because processing cannot begin until the whole packet has arrived, there may be benefit from reversion to a distributed checksum or parity detection code, similar to the way it was done in parallel PCI.

It is unclear at this stage whether split completions can be avoided in an optical fabric, but this situation may become clearer as the research progresses.

Results from this experiment should be apparent by Christmas ’08.

6.2 Experiment 2a:
Evaluate impact of latency

Martin et al. showed [MVCA97] that system overhead (of which latency is a component) is the factor with the greatest impact on performance. The object of this experiment is to see whether the latency attributable to buffering and interconnect matters to peripheral performance.

As observed in [MVCA97], overhead cannot readily be reduced in a real system. Instead, the effect of increasing overhead by use of an electronic delay line can be used to extrapolate what might happen if overhead were decreased past that of ordinary performance.

Performance will be measured using benchmarks appropriate to the application hardware connected to the apparatus as latency is gradually increased. For example, SpecSFS can be used to measure the performance of a network and a RAID controller.

A positive result shows that latency affects peripheral performance, and by inference, reduction in latency (such as by removing buffers from the interconnect) would improve performance.

There are good reasons to expect a positive result. [MVCA97, LCW+03, LWP04, Thi08] all suggest that HPC and financial applications are already sensitive to delays in the order of microseconds. If the result is negative or equivocal, it may be that the application examined isn’t sensitive enough to latency to show an effect. In this case, some investigation will be made into why the application isn’t significantly affected by latency, and if the hardware is available, to repeat the experiment with a more applicable workload.

If a case cannot be made, theoretically or experimentally, for the significance of interconnect latency, the focus of the research will switch to providing bandwidth past that of which an electronic interconnect would be capable.

This experiment involves designing some custom hardware to break out a one lane PCI Express slot to SMA connectors. One end will be a small PCI Express card with some hardware to replicate the reference clock for each of the devices connected. The other end will contain a PCI Express socket, a power connector and a regulator to supply the 3.3 volt rail to the socket.

The experiment will be implemented on a Xilinx Virtex prototype board interposed between these break-out boards. Allow about four months for this experiment.
6.3 Experiment 2b: Evaluate slot size selection

For the sake of the manageability of fabric scheduling, and given the lack of an optical equivalent to RAM, access to an optical switch fabric is typically divided into time slots [JRG+03].

A trade-off must therefore be made between fabric utilisation, overhead, and latency. For sufficiently large time slots (where slot length >> MTU, such as in optical burst switching [JV05]), efficiency is traded for latency. The results of experiments 1 and 2a will indicate whether this is a viable trade-off. For slot length ≈ MTU, latency is improved at the expense of overhead.

This experiment is designed to investigate the effect of quantising transactions at set intervals. The apparatus used for experiment 2a can be retasked to forward packets according to a given time slot size. As in the last experiment, a benchmark will be used to evaluate the effect of different time slot sizes on various applications.

The design of the fabric access scheduler will depend on the results of this experiment.

The apparatus required for this experiment is the same as for 2a. All that should be required is altering the way transactions are emitted from the delay line. Allow one month for this.

6.4 Experiment 3: Evaluate an optical fabric

This experiment seeks to build an electronic model of a switch fabric. It is essentially a PCI Express bridge with two subordinate buses. The hardware required is much the same as for Experiment 2, with the addition of a second card adapter.

The FPGA on the Virtex prototype board will contain a partial PCI Express end-point for each device connected, and the model of the optical switch itself. Although electronic, the model will be designed subject to the same rules and limitations of an optical implementation.

PCI Express is designed to expect buffers on both directions and ends of a link, and uses credit advertisements to mediate flow control between the buffers.

It expects to be able to transmit at any time there is sufficient credit to do so, and thus it does not have to arbitrate for access to the fabric. Conversely, in a time division multiplexed switch fabric, an agent can only transmit when it is in possession of an open channel (whether the channel is arbitrated for, or the agent is allocated a dedicated time slot.)

In order to interface with the architecture being developed here, the credit based flow control mechanism of PCI Express will need to be replaced with an arbitration mechanism. The end-point module will make this conversion along with any other optimisations of the protocol (such as distributed error detection codes). The latency introduced by this step will have to be allowed for when calculating the efficiency of the electronic model as a whole.

As in a real optical implementation, the bandwidth of each of the downlink ports will be the same as the bandwidth of the uplink port. When multiple downlink agents are active, the uplink bandwidth is shared between the active downlink agents, and no one downlink can reach 100% utilisation.
Unlike experiment 2, PCI Express is now being used as a transport emulating another protocol. Performance of even a single active downlink agent cannot be expected to equal the performance of that same agent when speaking native PCI Express. Evaluation of the performance of the interconnect cannot therefore be done with benchmarks as in previous experiments because end-to-end measurements will include delay components that don’t belong to the model (such as the PCI Express links themselves.) Instead, the model and end-point adapters will have to be instrumented to measure parameters such as fabric utilisation and arbitration latency. It might be possible to measure true round-return latency differentially, using the same technique as in experiment 1. Useful data could also be collected if a “native” agent were designed and implemented inside the prototype board.

This is a challenging and complex experiment to execute, and will require the design and implementation of a large number of new components including fabric model, scheduler, the aforementioned end-point, and enough logic to implement a simple PCI bridge. This might take six months.

If successful, this experiment will have produced a demonstrator which shows the feasibility of an optical interconnect.

6.5 Experiment 4: Optical implementation

The electronic model can model architectural sources of latency, but it cannot readily model technological parameters like the time taken to recover a new clock after the fabric has been reconfigured. CDR latency has the potential to be a major factor in the overall overhead associated with the optical switch. A possible fourth experiment therefore involves building a real optical switch to investigate practical matters like CDR latency.

The apparatus is an extension of Experiment 3. It will require the use of a Virtex evaluation board which can accommodate six high-speed serial ports (enough for three PCI Express links and three optical transceivers), four SOAs and associated optical couplers and current drivers.

![Figure 3: Block diagram of components in Experiment 4.](image)

7 Related issues

7.1 Power consumption and density

Since any current implementation will be built using discrete components, to a great extent density (the physical volume occupied by the switch) - and to a lesser extent, its power consumption - is not something that can readily be analysed in this thesis. Ultimately, density and power consumption will depend on the degree of integration which can be achieved (and which will also affect cost).

7.2 Scheduling

Until a viable optical equivalent to RAM becomes available, switch fabrics of the kind mentioned here aren’t random access and therefore require a scheduler to arbitrate access between devices. Aside from slot size, the design of that scheduler is a critical factor in the latency through the switch.

PCI Express divides traffic into two classes: link layer and transaction layer. Transaction layer pack-
ets carry data transfers (PCI transactions), while link layer packets are responsible for flow control and signalling errors (amongst other things.) PCI Express is also a windowed protocol, meaning that within sensible limits, the timing of acknowledgements affects only the amount of buffering required within an endpoint, not throughput.

The scheduler can be designed to take advantage of the features of the protocol it is transporting, and of the switching technology in which the fabric is implemented - such as by pipelining (advance scheduling) of multiple requests and delaying acknowledgements to an endpoint until the fabric is next configured to transfer more data with that endpoint.

7.3 Subsequent work

The results of these experiments will determine where the work goes next. Problems encountered will inspire research into how to solve them, and successes will provide opportunities to study ways of leveraging the technology.

The arbitration and time slot mechanism is an area which might benefit from further research, since both are significant factors in the latency of the switch. Analysis of power consumption is worth doing.

CPU and memory interconnects are not covered by this research, and would no doubt benefit greatly from reduced latency and power consumption. This is probably the most promising and enabling line of inquiry.

8 Conclusion

The principle contribution of this research is not whether a photonic switch can be developed, but whether it can be done efficiently enough to make it both a viable replacement for existing PCI Express and to provide scope to address the considerations outlined in the introduction.

Despite the considerable advances in photonic components, the field is still in its infancy compared with what we can do with electronics today. It is clear, even at this point, that for optical interconnects to become a commercial reality in the future, much work will need to be done both in the development and integration of the technology, and in understanding the implications for computer architecture of such systems.

9 Proposed thesis chapters

I Introduction

II Background

There’s a good number of papers which look relevant at the up-coming Hot Interconnects 16 conference, which will be worth mentioning here.

1 Evolution of power and performance in local interconnects

Bring in experimental results from the latency work.

2 Existing photonic switches.

Cover IP routers, SWIFT and SOAPS from Intel, Rod Tucker’s analyses on power and density of optical IP routers vs. electronic IP routers.

3 Comparison between PCI Express and Infiniband

HotI 16, and any other work that comes up.

4 Applications for an optical interconnect

I’ve been focusing on PCI Express, but also mention CPU interconnects etc.

5 On-chip and off-chip optical interconnects

Multi-drop optical buses from HotI 16; waveguides in silicon (NoC applications etc) and deposited on printed circuit boards, plus the recent work from CUDOS (photonic integration, scratch waveguides)

III Architectural considerations

1 Switch fabric configuration

   i Overview of switch design

Very brief mention of choices in interconnect network architecture
2 Overview of photonic components
Briefly describe the choices, and why we select what we have

i Transceivers
Lasers and photodiodes, VCSELs for direct coupling to PCBs, CDR

ii Amplifiers and switching elements
SOAs and YDF A; trade offs in switching time and noise; MEMS; Lithium niobate Mach-Zehnder modulators

iii Passives
Couplers and splitters, insertion loss and associated impact on the depth of a switch fabric

iv Loss and noise
Extent to which loss in passives can be compensated for with amplifiers, with reference to noise

v Inter- and intra-chip Waveguides
Fibres; on-chip and off-chip waveguides (a bit more detail than mentioned in associated work)

3 Medium access arbitration

i Time slots and their structure
Guard bands for switching, CDR and data time. CDR probably the limiting factor, so comment on solutions

ii The role of buffering
Use of buffering in conventional non-blocking switch fabrics; bufferless switches; cost of OEO buffering; use of edge buffering

iii Scheduling
a Overview of algorithms
Algorithms as appropriate to interconnect type

b Transaction types and scheduling
Pipelining of consecutive requests; delayed acknowledgements and impact on edge buffering requirements
1 *Throughput and latency performance*

2 *Power performance*

3 **Analysis of time slot allocation**
   - **Clock recovery time**
     Analysis of how significant CDR performance was to overall performance
   - **End-point buffering requirements**
     Analysis of the effect of time slots on end point buffering requirements

4 **Analysis of medium access scheduler**

5 **Synchronous endpoints**
   Analysis of how the implicit requirement that even low performance end-points would have to run at memory/CPU interface speed may be ameliorated.

VII **Conclusion**

**References**


