A 2.5Tb/s LCS Switch Core

Nick McKeown
Costas Calamvokis
Shang-tse Chuang
Outline

LCS: Linecard to Switch Protocol
  ❖ What is it, and why use it?

2. Overview of 2.5Tb/s switch.

3. How to build scalable crossbars.

Next-Generation Carrier Class Switches/Routers

Switch Core

Up to 1000ft

Linecards

August 20th, 2001
Benefits of LCS Protocol

1. Large Number of Ports.
   - Separation enables large number of ports in multiple racks.
   - Distributes system power.

2. Protection of end-user investment.
   - Future-proof linecards.

3. In-service upgrades.
   - Replace switch or linecards without service interruption.

4. Enables Differentiation/Intelligence on Linecard.
   - Switch core can be bufferless and lossless. QoS, discard etc. performed on linecard.

5. Redundancy and Fault-Tolerance.
   - Full redundancy between switches to eliminate downtime.
Main LCS Characteristics

1. Credit-based flow control
   - Enables separation.
   - Enables bufferless switch core.

2. Label-based multicast
   - Enables scaling to larger switch cores.

3. Protection
   - CRC protection.
   - Tolerant to loss of requests and data.

4. Operates over different media
   - Optical fiber,
   - Coaxial cable, and
   - Backplane traces.

5. Adapts to different fiber, cable or trace lengths
LCS Ingress Flow control

1: Req

2: Grant/credit

3: Data

1: LCS Ingress Flow control

Linecard

Switch Port

Switch Fabric

Switch Scheduler

Seq num
LCS Adapting to Different Cable Lengths
LCS Over Optical Fiber
10Gb/s Linecards

10Gb/s Linecard

2.5Gb/s LVDS

GENET Quad Serdes

12 multimode fibers

12 multimode fibers

10Gb/s Switch Port

Switch Fabric

Switch Scheduler
Example of OC192c LCS Port

LCS Protocol to OC192 Linecard

12 Serdes Channels
Outline

1. LCS: Linecard to Switch Protocol
   - What is it, and why use it?
2. Overview of 2.5Tb/s switch.
3. How to build scalable crossbars.
Main Features of Switch Core

2.5Tb/s single-stage crossbar switch core with centralized arbitration and external LCS interface.

1. Number of linecards:
   - 10G/OC192c linecards: 256
   - 2.5G/OC48c linecards: 1024
   - 40G/OC768c linecards: 64

2. LCS (Linecard to Switch Protocol):
   - Distance from line card to switch: 0-1000ft.
   - Payload size: 76+8B.
   - Payload duration: 36ns.
   - Optical physical layers: 12 x 2.5Gb/s.

3. Service Classes: 4 best-effort + TDM.

4. Unicast: True maximal size matching.

5. Multicast: Highly efficient fanout splitting.

6. Internal Redundancy: 1:N.
2.56Tb/s IP router

1000ft/300m

Port #1

Port #256

2.56Tb/s switch core

LCS

Linecards
Switch core architecture

Port #1
- Cell Data
- LCS Protocol
- Request
- Grant/Credit

Port Processor

Scheduler

Crossbar

Port #256
- LCS Protocol
- Cell Data
- LCS Protocol

Port Processor

optics
Outline

1. LCS: Linecard to Switch Protocol
   - What is it, and why use it?

2. Overview of 2.5Tb/s switch.

3. How to build scalable crossbars.

How to build a scalable crossbar

1. Increasing the data rate per port
   - Use bit-slicing (e.g. Tiny Tera).

2. Increasing the number of ports
   - Conventional wisdom: $N^2$ crosspoints per chip is a problem,
   - In practice: Today, crossbar chip capacity is limited by I/Os.
   - It’s not easy to build a crossbar from multiple chips.
Scaling: Trying to build a crossbar from multiple chips

Building Block:

16x16 crossbar switch:

Eight inputs and eight outputs required!
Scaling using “interchanging”

4x4 Example

Cell time

Reconfigure every cell time

Reconfigure every half cell time

2x4 (2 I/Os)

2x4 (2 I/Os)
2.56 Tb/s Crossbar operation

Interchanger

2x2 fixed “TDM”

Crossbar A

128 x 256 xbar

Crossbar B

128 inputs

128 outputs

Interchanger

2x2 fixed “TDM”
Outline

1. LCS: Linecard to Switch Protocol
   - What is it, and why use it?
2. Overview of 2.5Tb/s switch.
3. How to build scalable crossbars.
How to build a centralized scheduler with true maximal matching?

Usual approaches

1. Use *sub*-maximal matching algorithms (e.g. iSLIP)
   - Problem: Reduced throughput.

2. Increase arbitration time: Load-balancing
   - Problem: Imbalance between layers leads to blocking and reduced throughput.

3. Increase arbitration time: Deeper pipeline
   - Problem: Usually involves out-of-date queue occupancy information, hence reduced throughput.
How to build a centralized scheduler with true maximal matching?

Our approach is to maintain high throughput by:

1. Using true maximal matching algorithm.
2. Using single centralized scheduler to avoid the blocking caused by load-balancing.
3. Using deep, strict-priority pipeline with up-to-date information.
Strict Priority Scheduler Pipeline

Scheduler Plane for
- 2.56Tb/s
- 1 priority
- Unicast and multicast
Strict Priority Scheduler Pipeline

P=0  P=0  P=0  P=0
P=0  P=0  P=0  P=0
P=0  P=0  P=0  P=0
P=0  P=0  P=0  P=0
P=0  P=0  P=0  P=0
P=0  P=0  P=0  P=0

0  1  2  3  4  5  6

Time

multicast

LCS Protocol

optics

Port Processor

Scheduler Priority p=0
Scheduler Priority p=1
Scheduler Priority p=2
Scheduler Priority p=3
Strict Priority Scheduler Pipeline

Why implement strict priorities in the switch core when the router needs to support such services as WRR or WFQ?

1. Providing these services is a Traffic Management (TM) function,
2. A TM can provide these services using a technique called Priority Modulation and a strict priority switch core.
Outline

1. LCS: Linecard to Switch Protocol
   - What is it, and why use it?
2. Overview of 2.5Tb/s switch.
3. How to build scalable crossbars.