Stardust

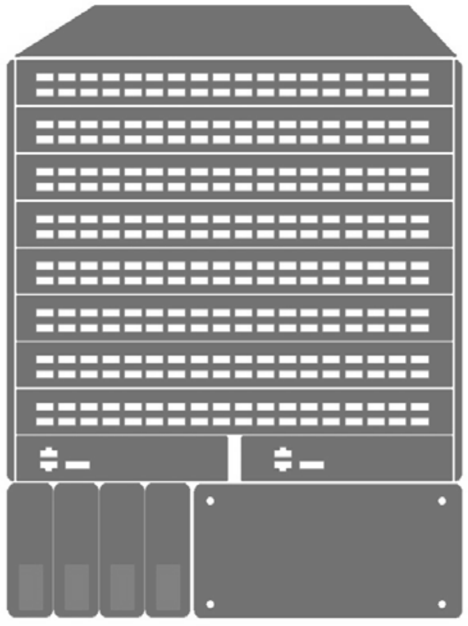
Divide and Conquer in the Data Center Network

Noa Zilberman
University of Cambridge

Golan Schzukin & Gabi Bracha
Broadcom

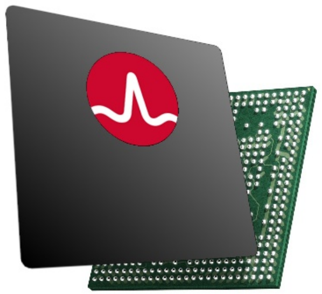
February 2019

Network switches



Switch chassis

Switch box



Switch silicon

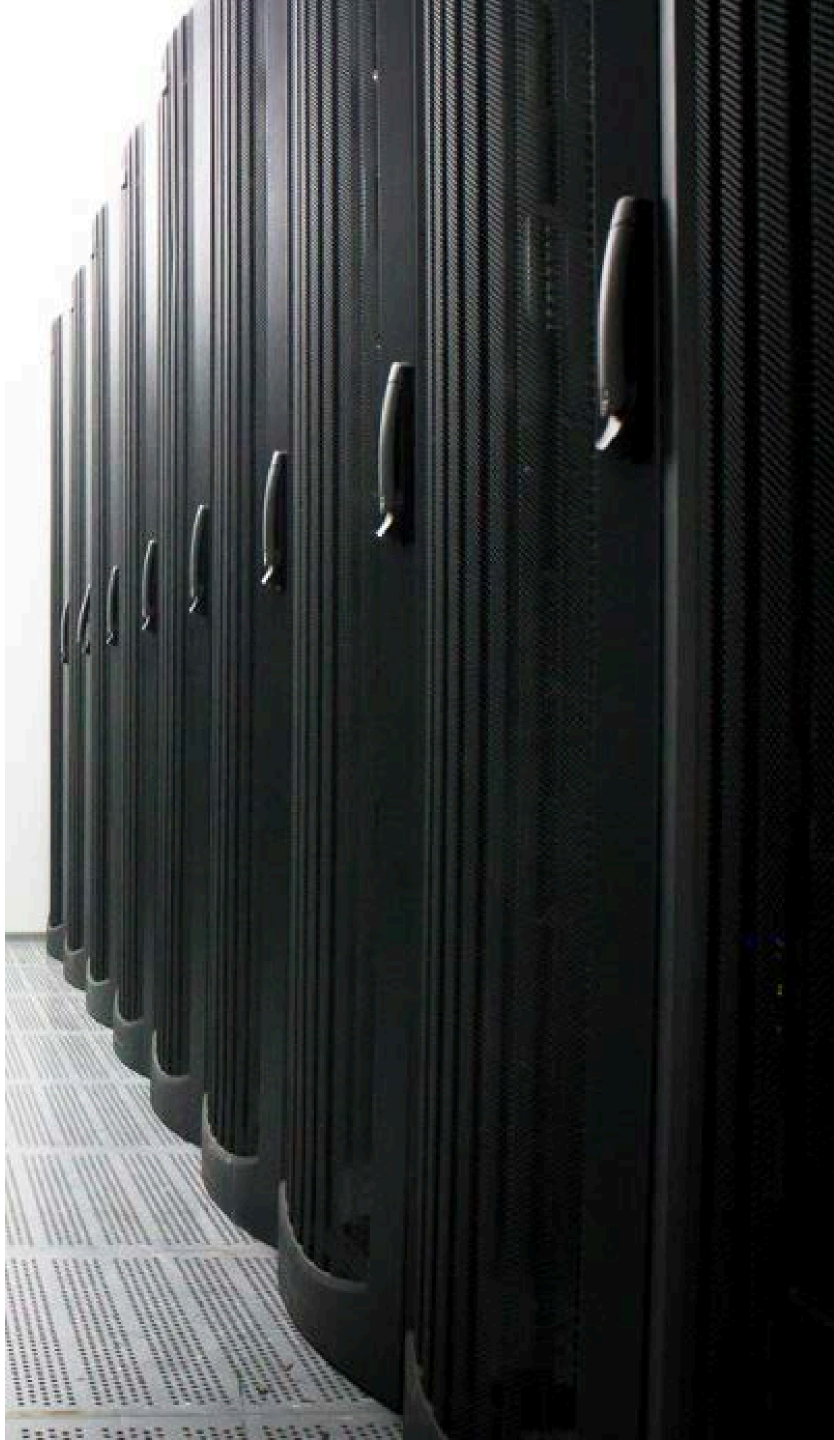
Scale: 12.8Tbps, 32×400GE

Network switch systems

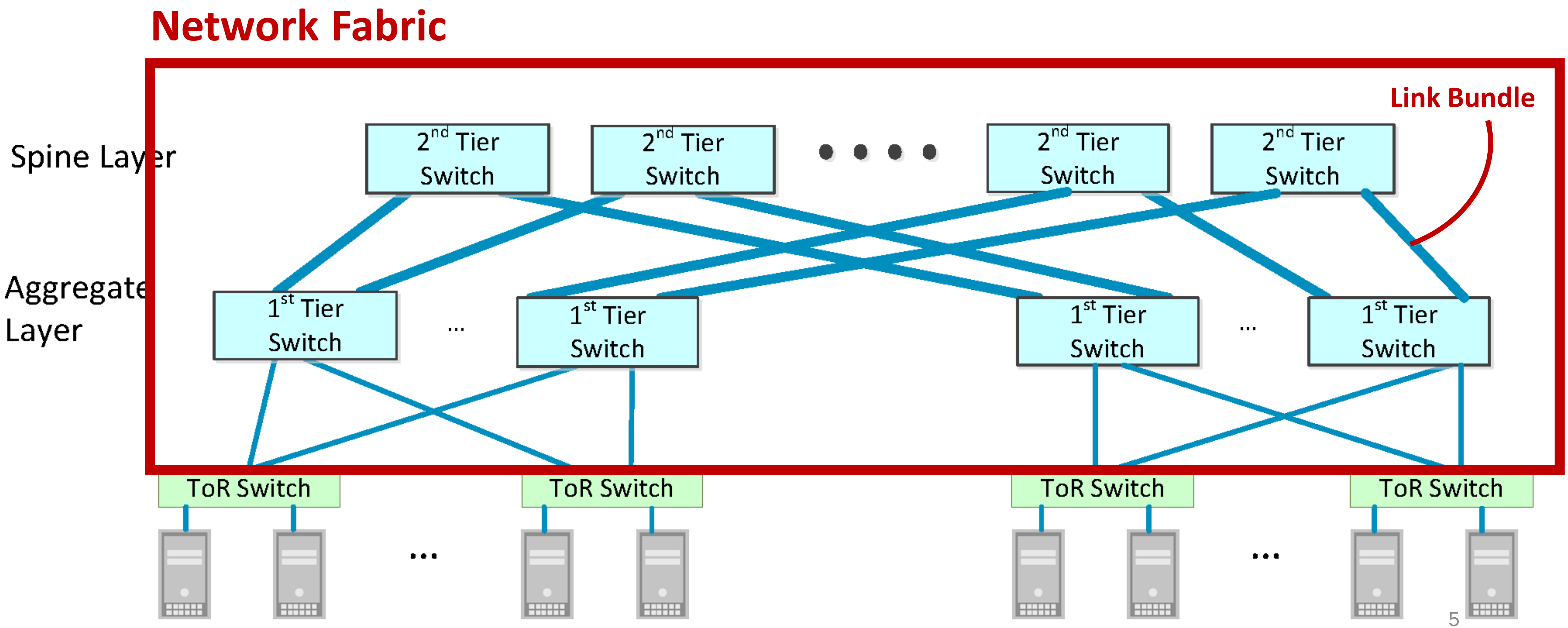
Scale: Petabit / second

Data center networks

Connecting 10K’s to 100K’s of servers



Do data center networks scale?



Do data center networks scale?

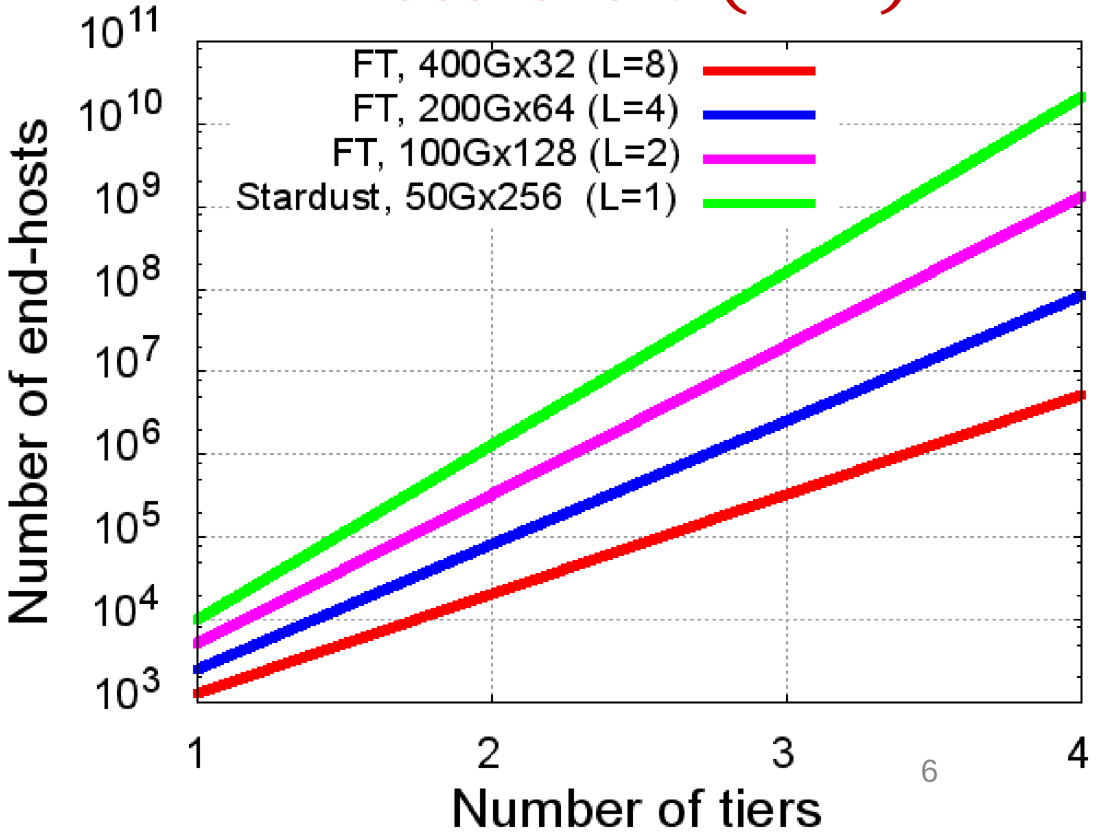
- Example: Building DC with 100K servers (2500 ToR switches)
- Option 1 – Link bundle of 1 (L=1):
  - 6.4Tbps Fabric Switch, 256×25G
  - Requires 2 Tiers

  #fabric-switches = 1172

- Option 2 – Link bundle of 4 (L=4):
  - 6.4Tbps Fabric Switch, 64×100G
  - Requires 3 Tiers

  #fabric-switches = 1954 (×1.66 more)

In a network of n tiers scale is $O(L^{-n})$



Do data center networks scale?

Observation:

A link bundle of one enables an optimum build of the network (i.e., less tiers, less switches, …)

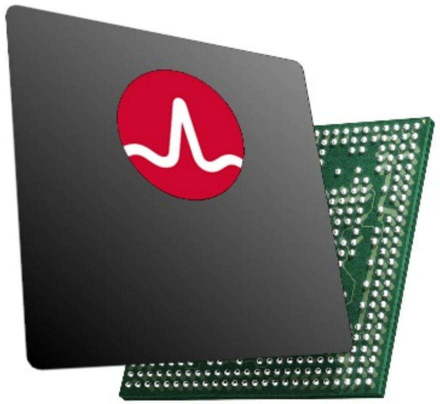
Designing new network devices

- A decade ago: *“Can we implement this feature?”*

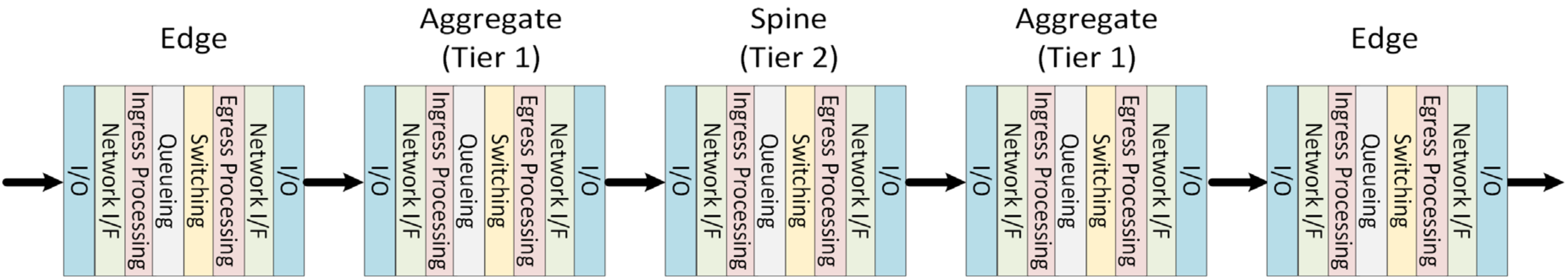
- Today: *“Is this feature worth implementing, given the design constraints?”*

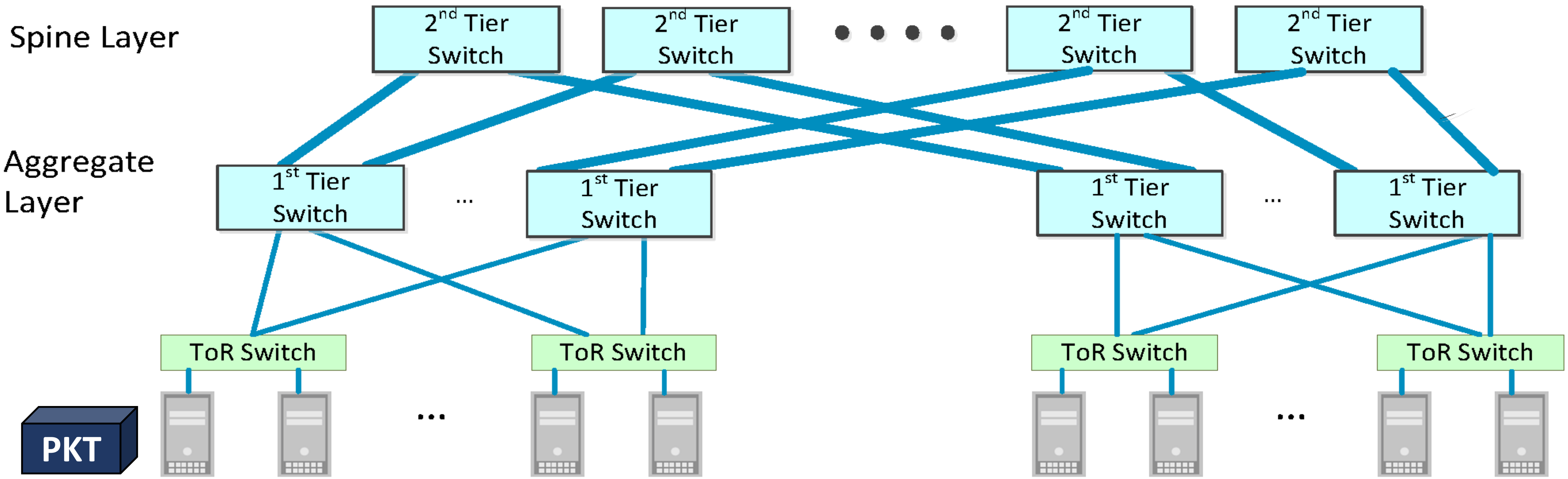
The resource wall

- Network silicon die > 7 Billion transistors (Tomahawk, 2014)
- Limited by:
  - Power density
  - Die size
  - Manufacturing feasibility

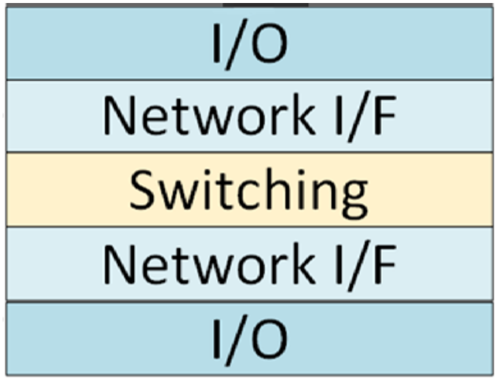


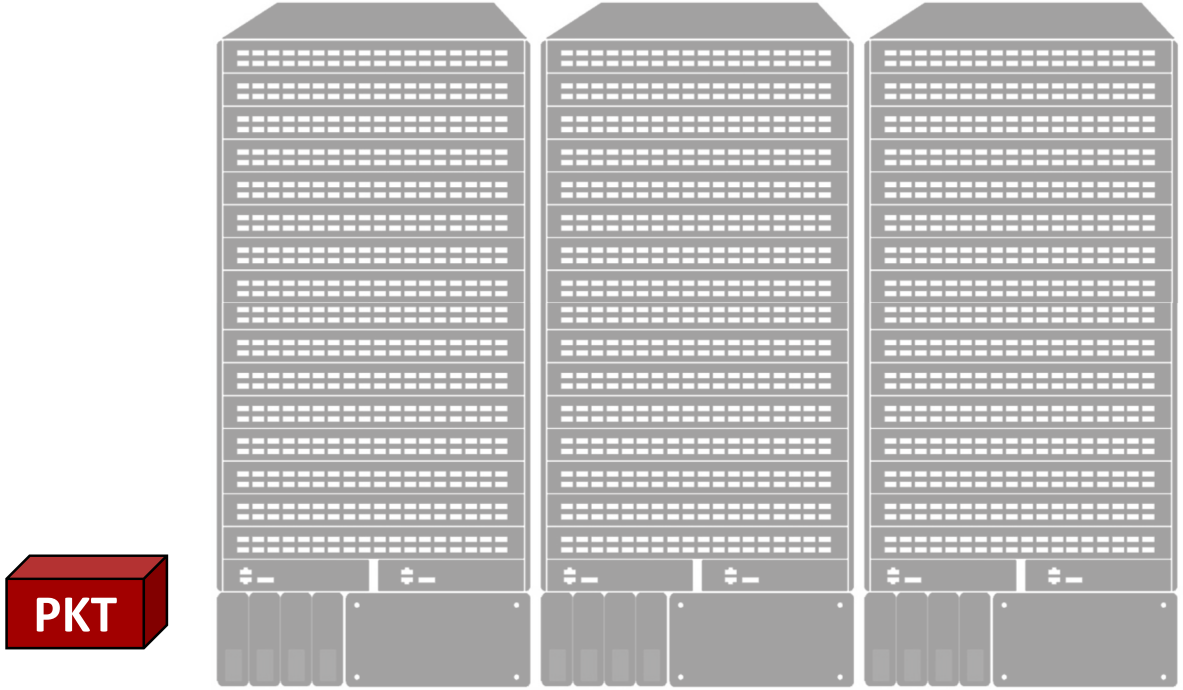
Data center network

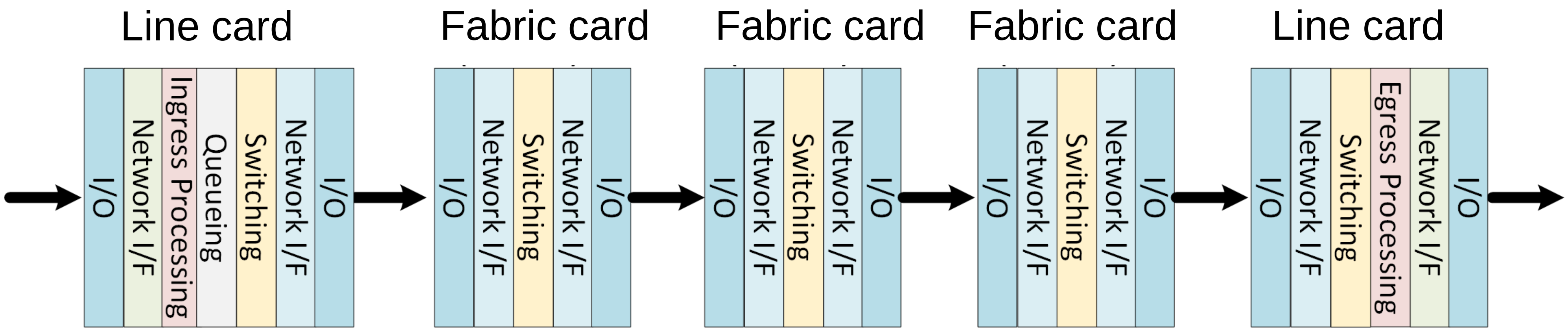




Switch system

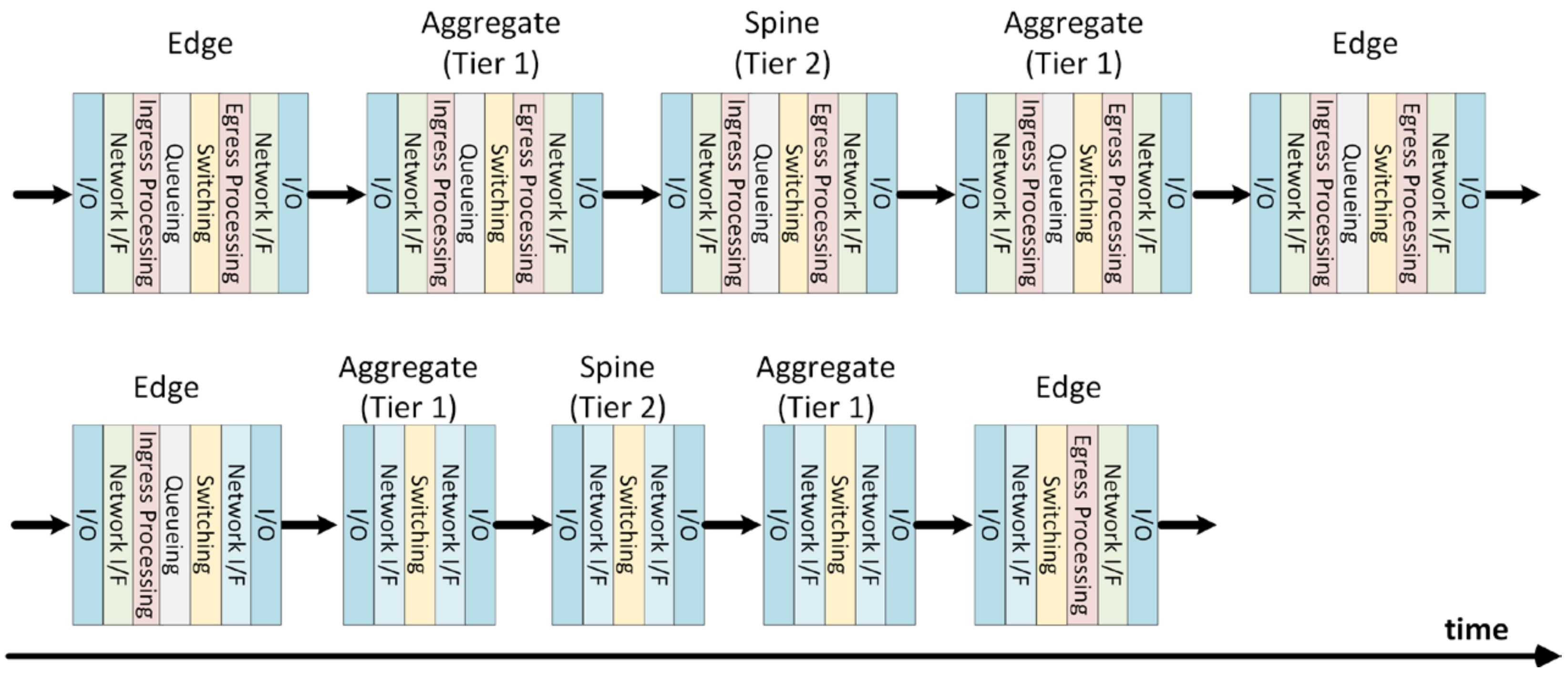






Why waste resources?
in *n* tier network

O(n×(Switching+2×I/O+2×NIF)+n×(Ingress Processing + Egress Processing + Queueing))

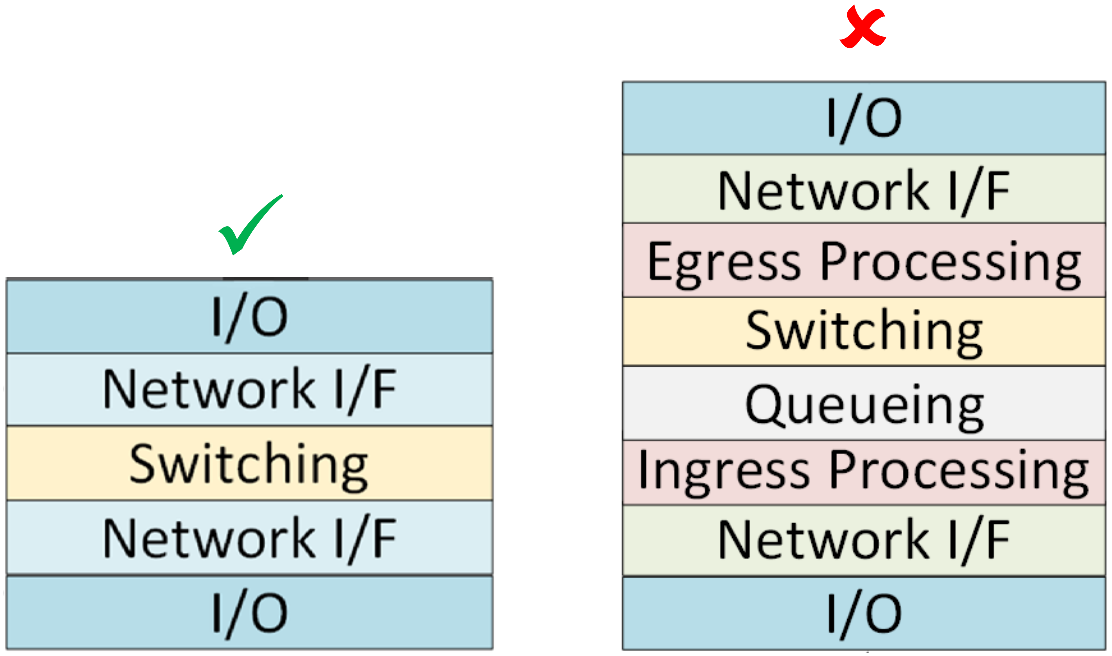


O(n×(Switching+2×I/O+2×NIF)+1×(Ingress Processing + Egress Processing + Queueing))

Why waste resources?

<u>Observation:</u>

Significant resources can be saved by simplifying the data center network



The single-pipeline switch

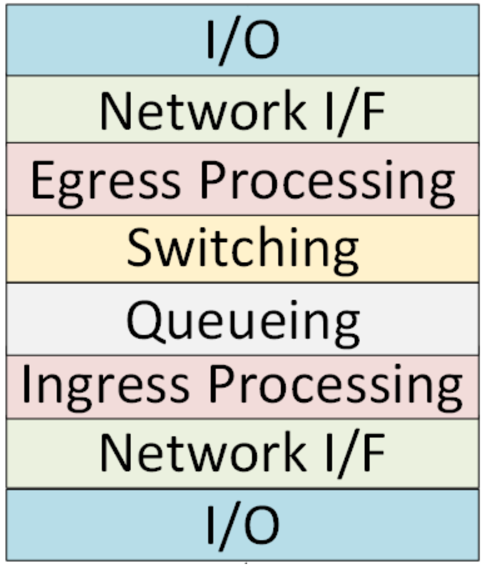
12.8Tbps Switches!

Lets convert to packet rate requirements:

5800 Mpps @ 256B (100GE→38.7Mpps)

19200 Mpps @ 64B (100GE→150Mpps)

But clock rate is only ~1GHz….



The single-pipeline switch

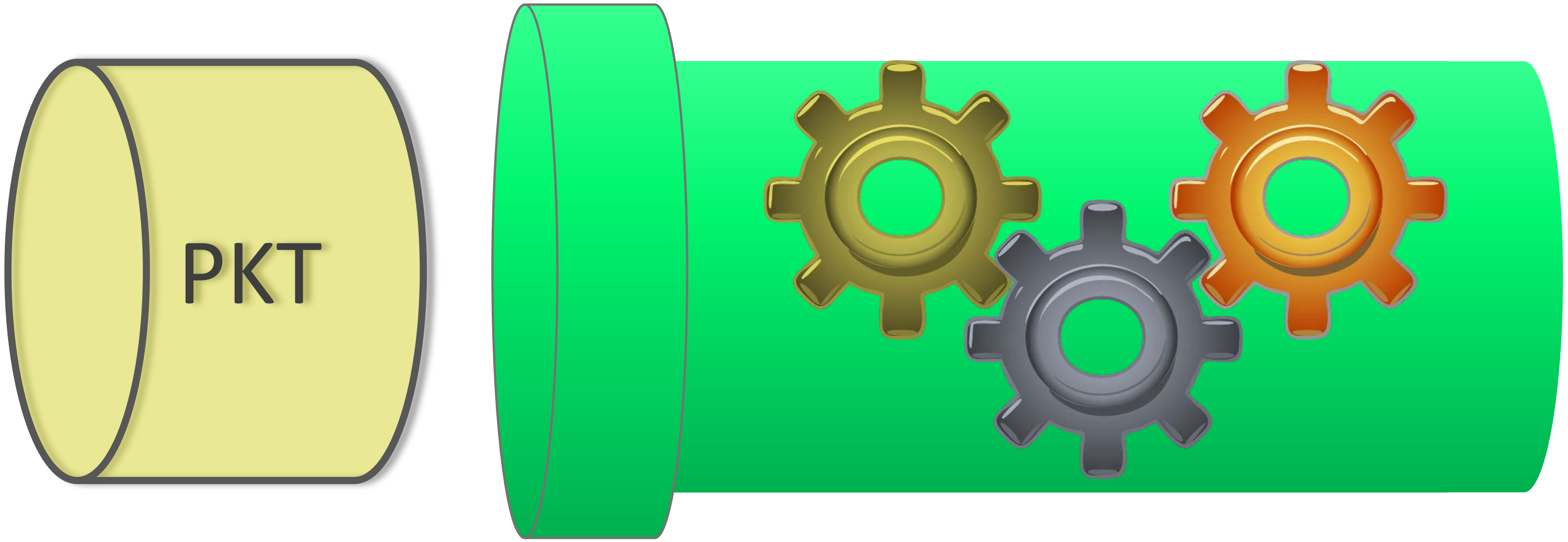
Observation:

To support full line rate for all packet sizes, network devices need to process multiple packets each and every clock cycle.

The age of multi core has reached switching…

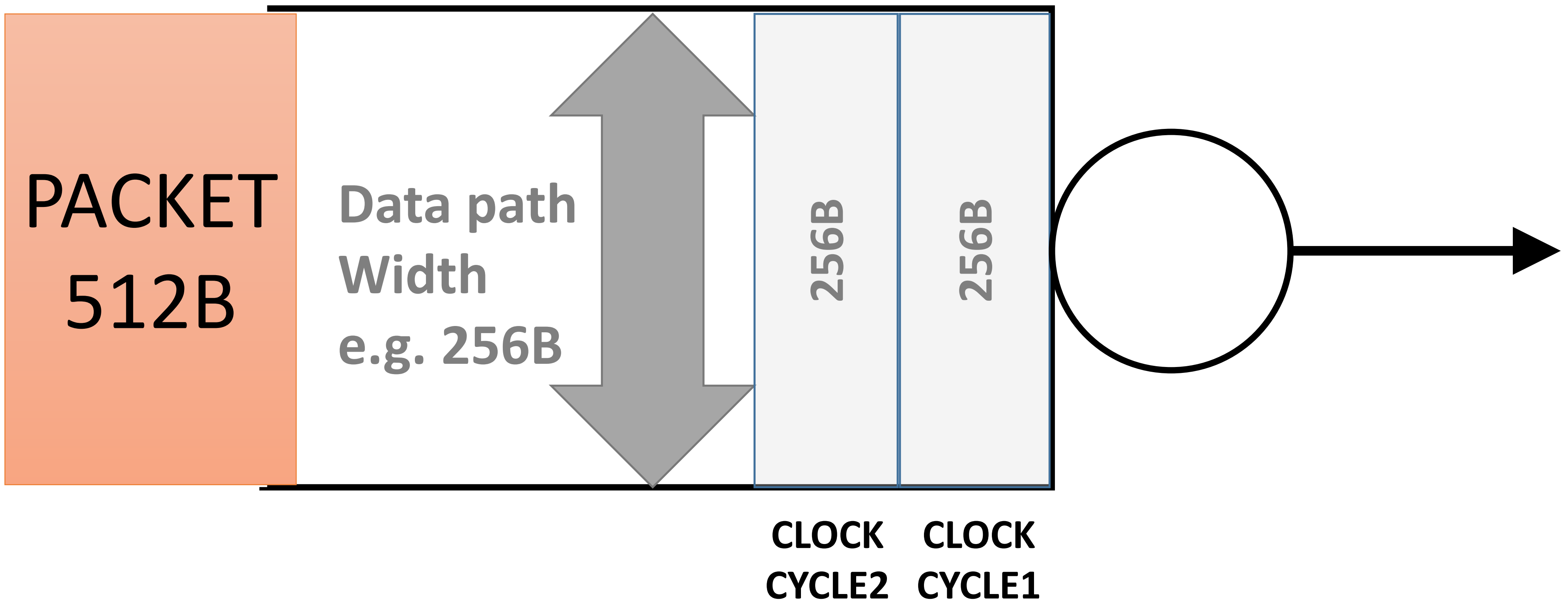
The switch pipeline

The common depiction:



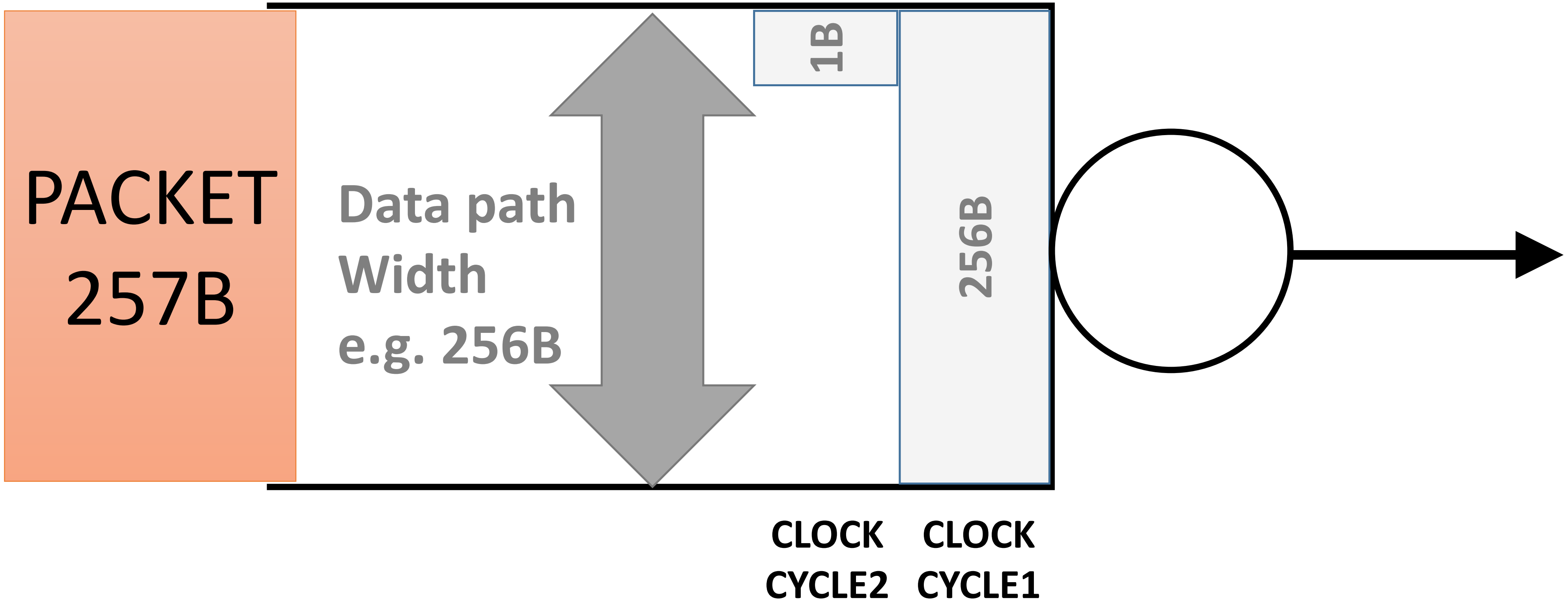
The switch pipeline

Actual Implementation:
Throughput = clock frequency x bus width

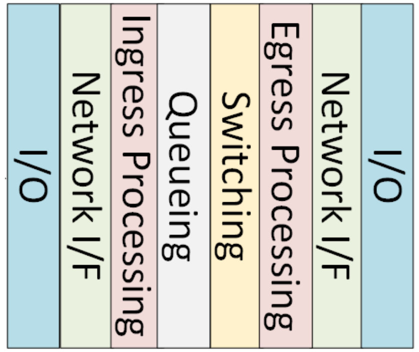


The switch pipeline

Actual Implementation:
Throughput ≠ clock frequency x bus width



The single-pipeline switch



12.8Tbps Switches!

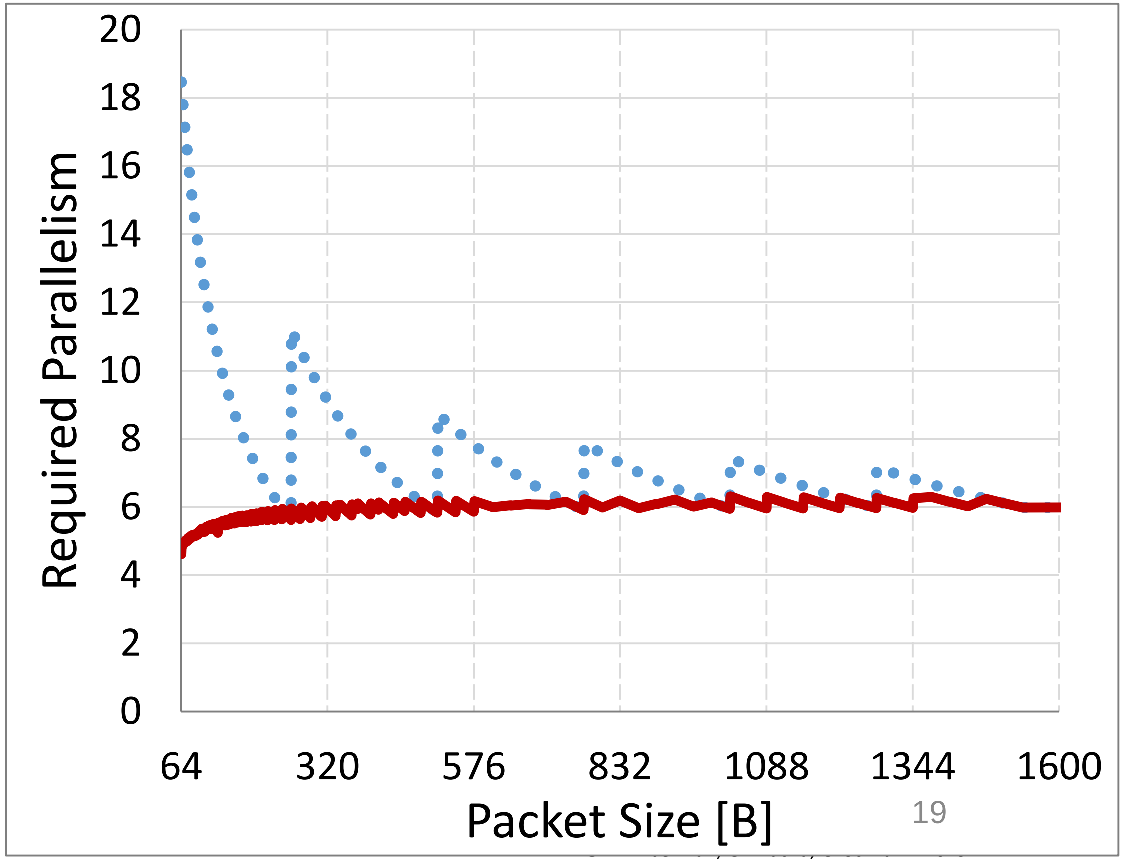
Lets convert to packet rate requirements:

5800 Mpps @ 256B (100GE→38.7Mpps)

19200 Mpps @ 64B (100GE→150Mpps)

But clock rate is only ~1GHz….

But if we pack data optimally…



The single-pipeline switch

Observation:

To support full line rate for all packet sizes, network devices need to process multiple packets each and every clock cycle.

Observation:

For best switch utilization, use fixed-size data units (cells)

The age of multi core has reached networking…

Observations

- A link bundle of one enables an optimum build of the network (i.e. less tiers, less switches, …)
- Significant resources can be saved by simplifying the network fabric
- To support full line rate for all packet sizes, network devices need to process multiple packets each and every clock cycle.
- For best switch utilization, use fixed-size data units (cells)

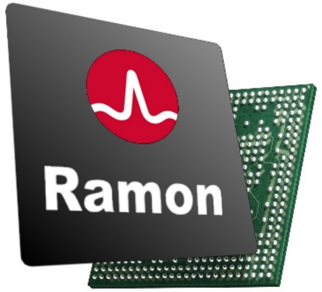
Introducing Stardust

From switch-system to data-center scale

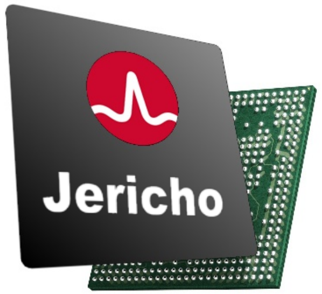
Introducing Stardust

Full details in our paper

- Complex edge, simple network fabric
- **Fabric Element** - Fabric device
  - A simple cell switch
- **Fabric Adapter** – Edge device
  - A packet switch
  - Quite similar to a ToR
  - Chops packets to cells



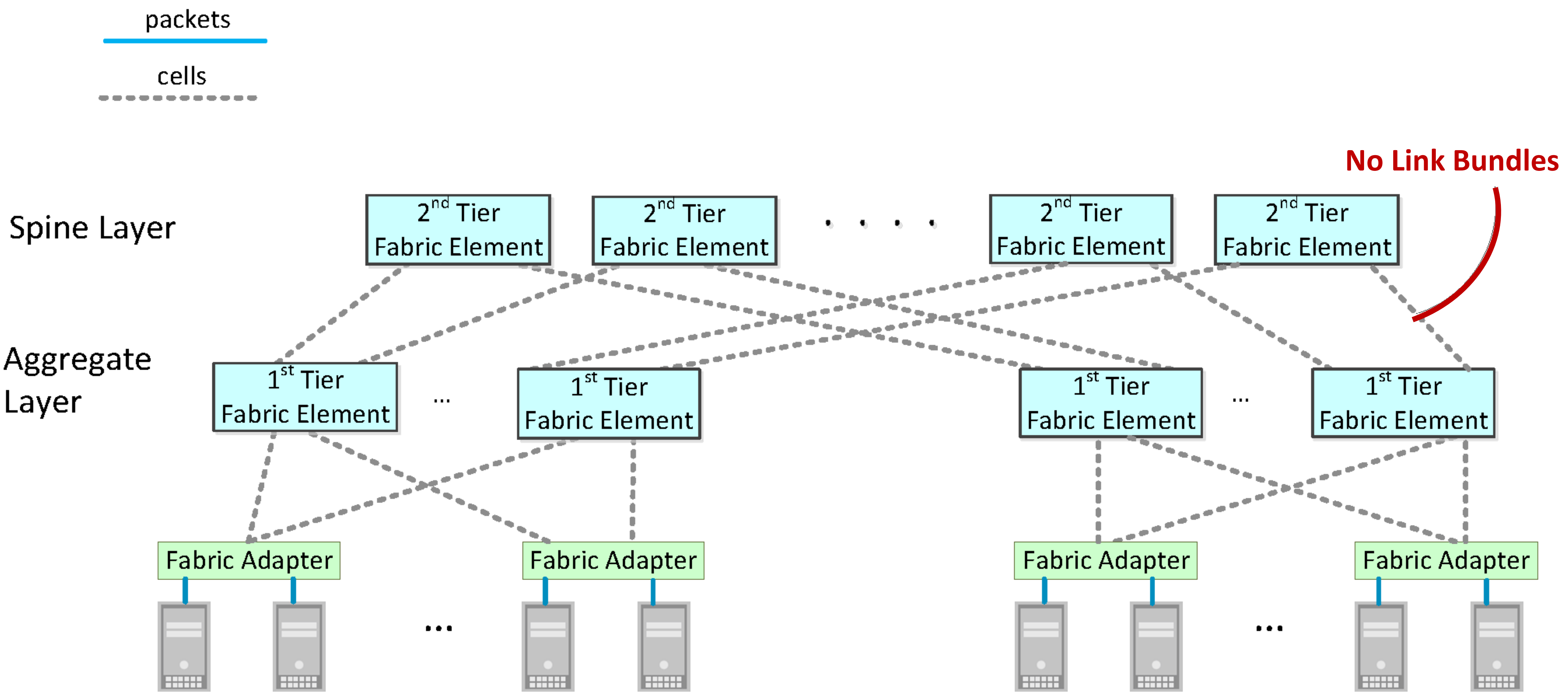
5th generation



7th generation

Widely used in switch-systems

A Stardust based network



Dynamic cell routing

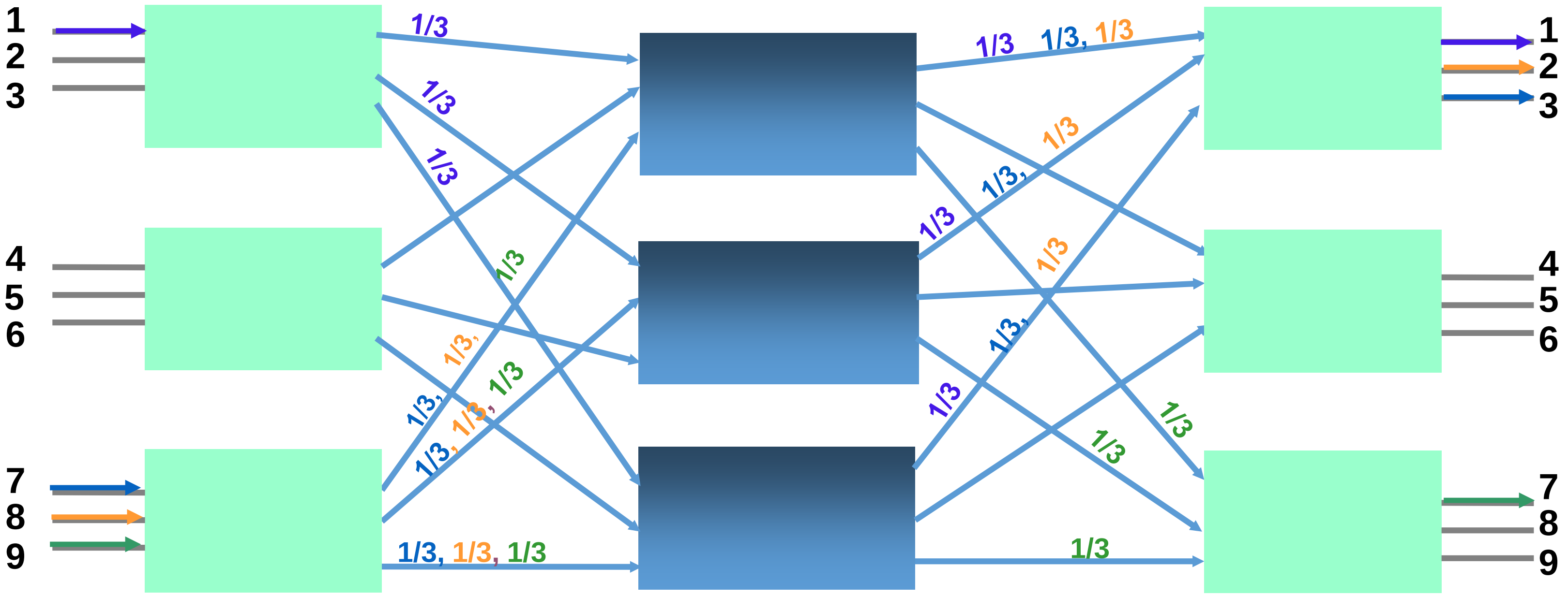
Input 1 → Output 1

Input 9 → Output 7

Input 8 → Output 2

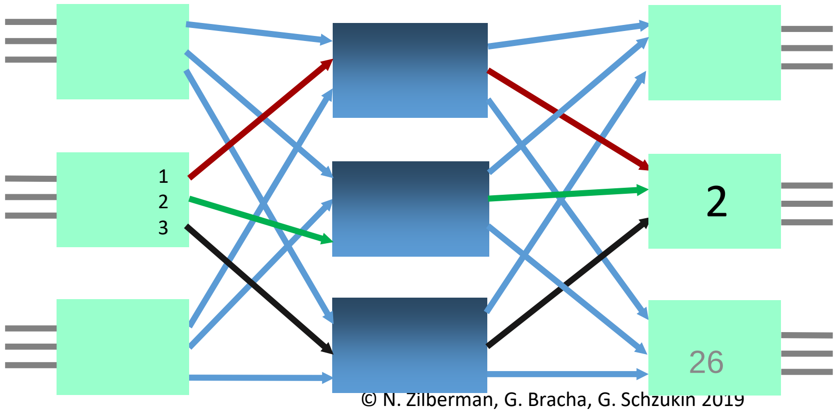
Input 7 → Output 1

→Non-Blocking



Reachability table

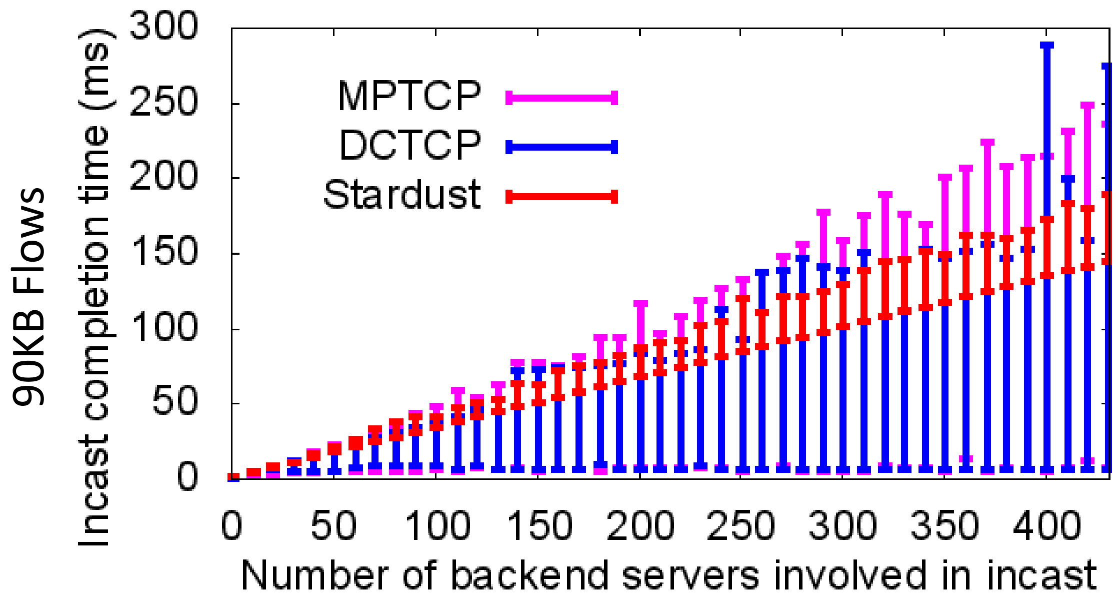
- Need to know only the destination Fabric Adapter
  - 1M virtual machines → 100K end hosts → 2500 Fabric Adapters
- Entries indicate “reachable through these links”
  - “You can get to Fabric Adapter 1 using links 1,5,8,14,36”
  - Bitmap of size *“switch radix”*
- Automatically constructed and updated
  - Using reachability messages



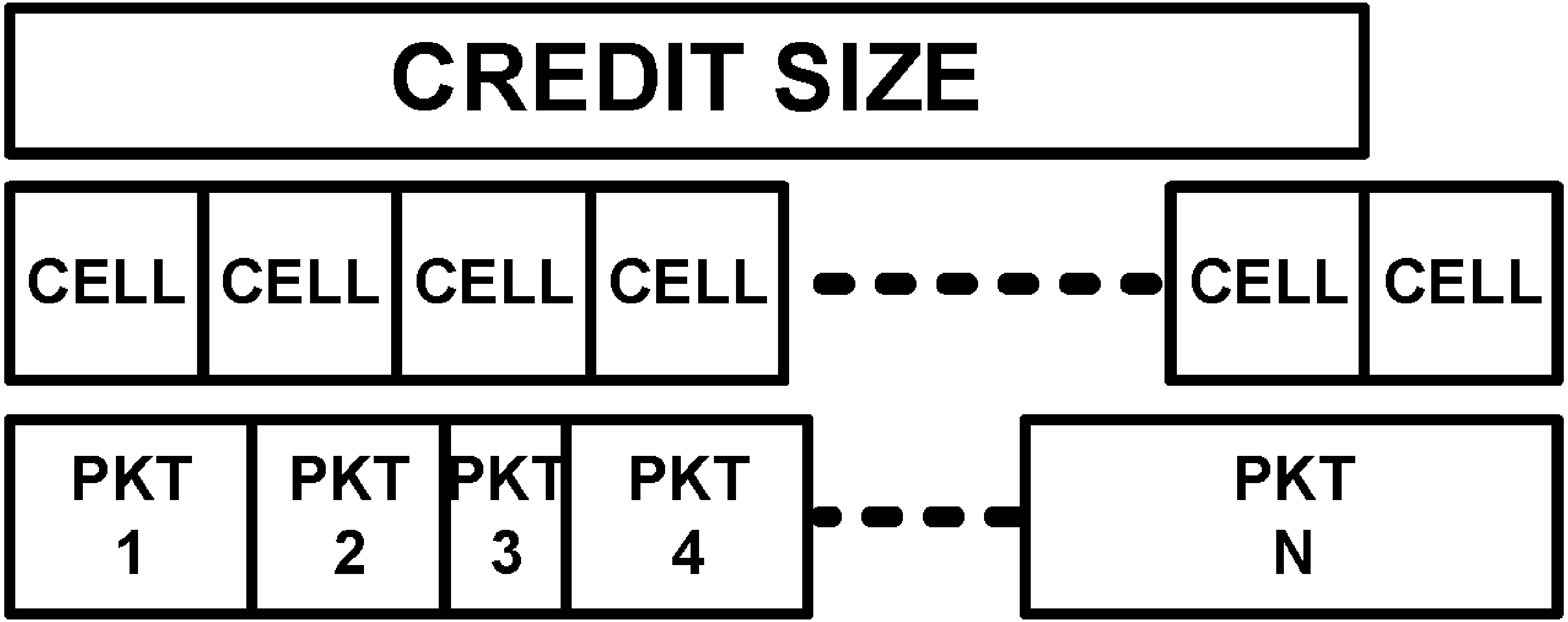
Buffering and scheduling

- Packet buffering at the edge
  - Using virtual output queues (VOQ) at the ingress Fabric Adapter
- A distributed scheduled fabric
  - A Fabric Adapter generates credits (e.g. 4KB) to all non-empty associated VOQ

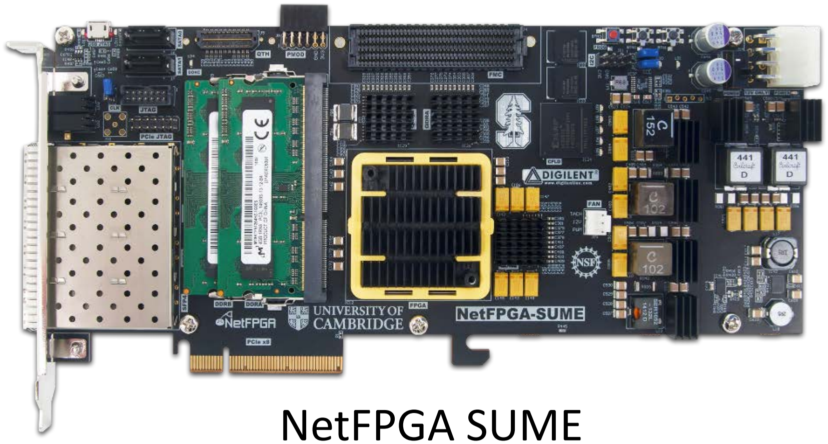
432-node Fat-Tree (simulation)



Packet packing

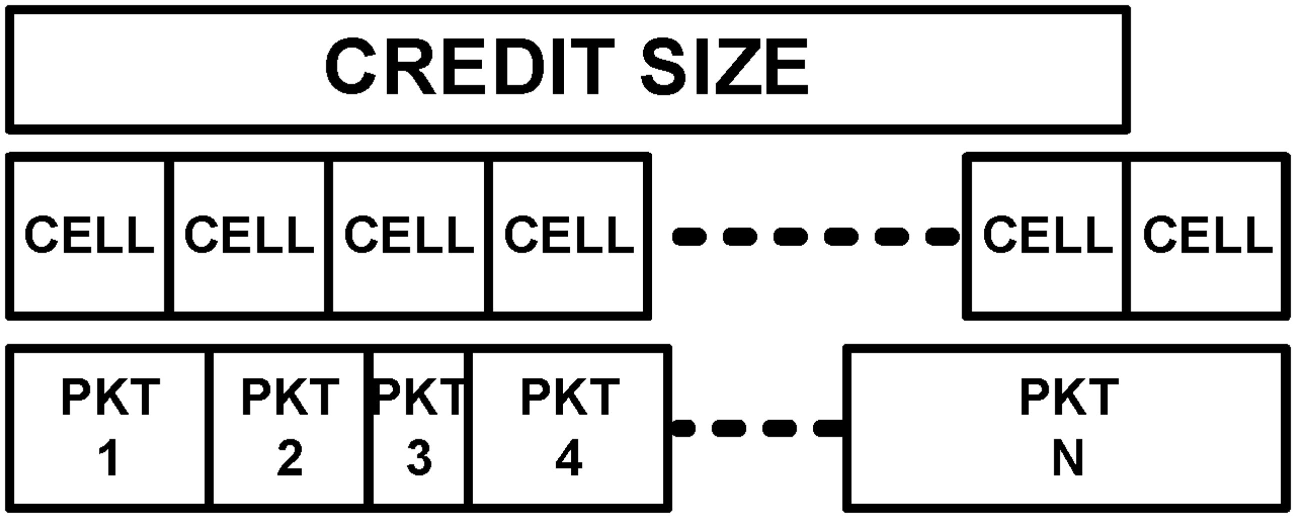


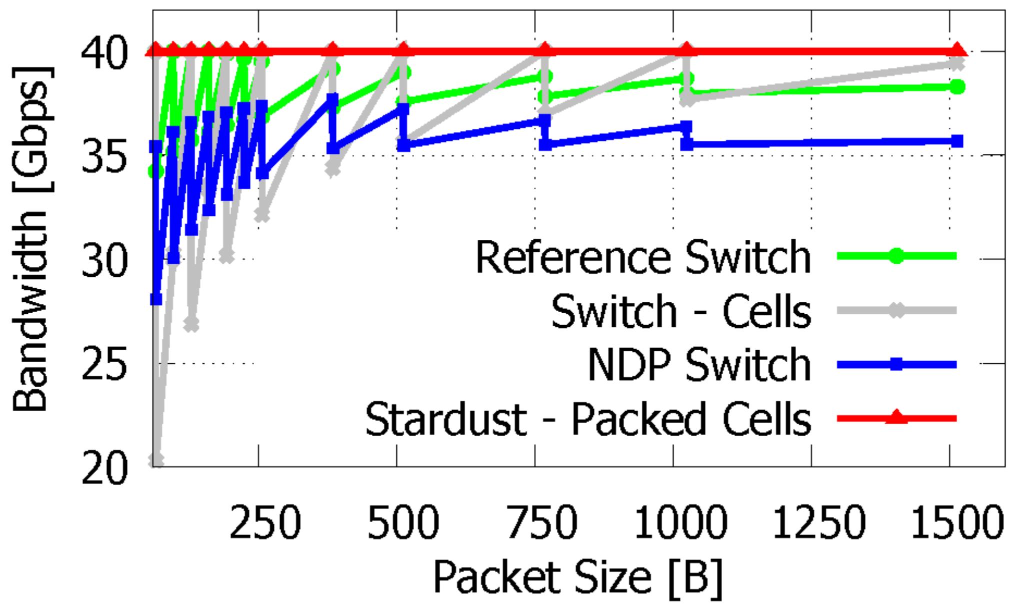
Packet packing

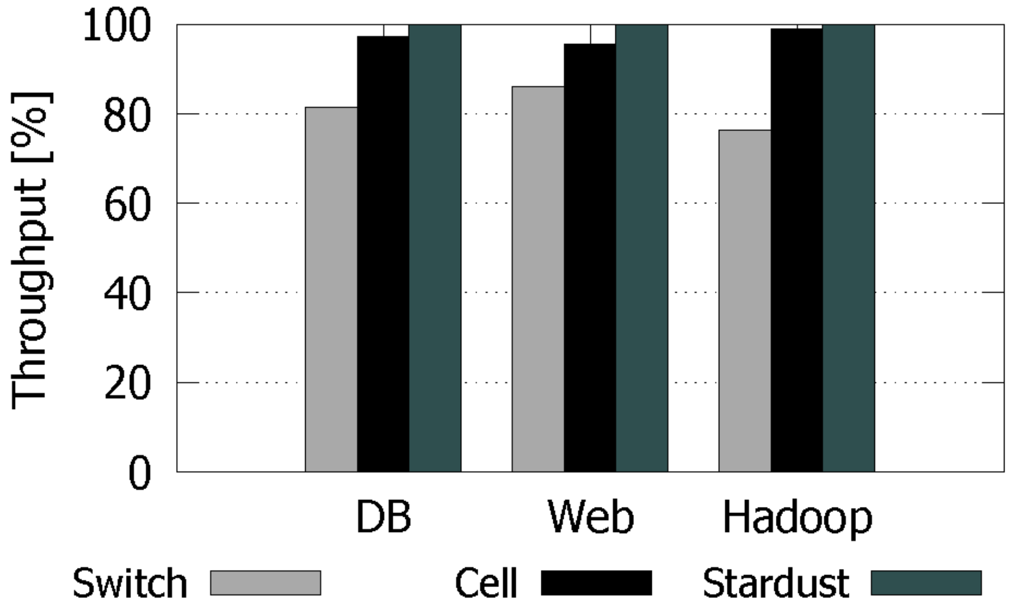


NetFPGA SUME

+







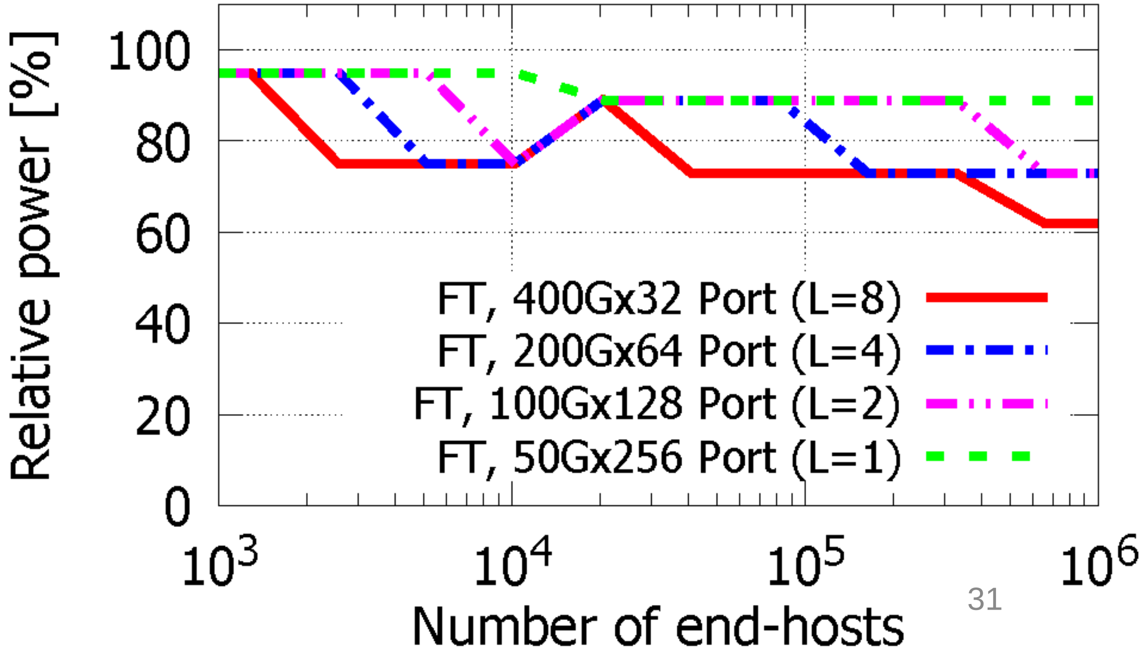
Properties

Integration matters!

- ✓ Protocol and traffic pattern agnosticism
  Cell switching & packing, dynamic routing, fabric scheduling
- ✓ Improved resilience and self healing
  Reachability messages, link bundling, dynamic routing
- ✓ Less network tiers, better scalability
  Link bundling, reachability messages, dynamic routing
- ✓ Optimal load balancing
  Dynamic routing, cell switching & packing, fabric scheduling
- ✓ Lossless transmission
  Fabric scheduling, dynamic routing, cell switching, reachability messages
- ✓ Incast absorption
  Fabric scheduling, dynamic routing, cell switching, reachability messages
- ✓ Pull fabric and port fairness
  Fabric scheduling, dynamic routing, cell switching, link bundling

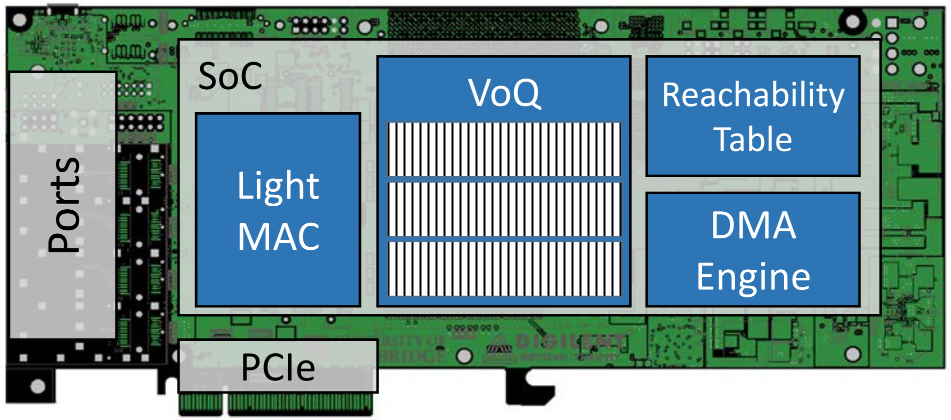
Power and cost – entire network

- Less network tiers → less devices
- Less power & area (cost) per device
  - Fabric Element saves 35% of power
  - Fabric Element saves 33.3% of silicon area
    - Save 87% of header processing area
    - Save 70% of network interface area



What about the future?

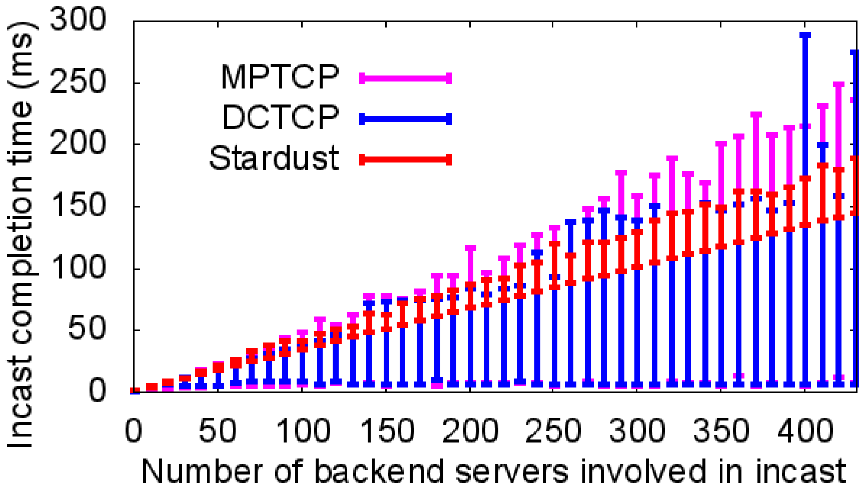
- Scalability of ToR / Fabric Adapter is the bottleneck
- Let us replace the **ToR** with a **Fabric Element**
- Let us turn the **NIC** into a **Fabric Adapter**
  - Lighter MAC
  - Smaller tables
  - Limited VOQs
  - Fabric adapters already support DMA



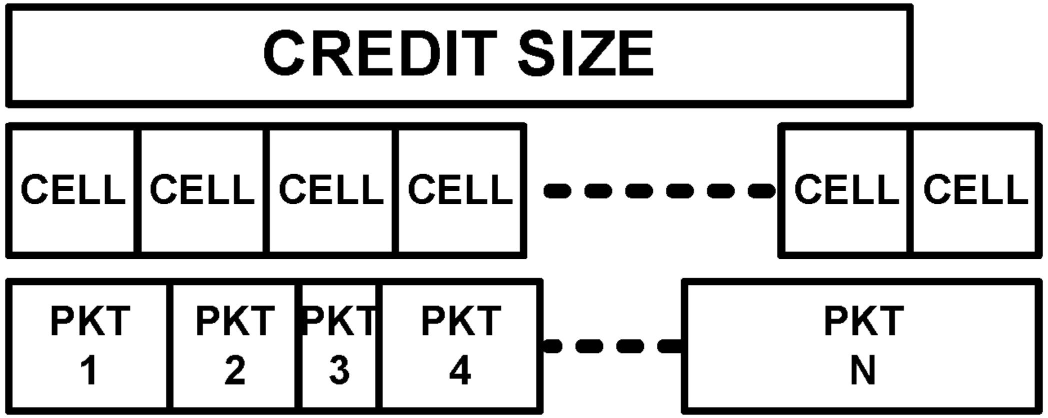
Stardust - summary

From switch-system to data center scale:

- Simple network fabric
- Push complexity to the edge



- Combines:
  - Cell switching and Packet packing
  - Load balancing
  - Scheduled fabric
  - Reduced network tiers
- Better performance
- Lower power, lower cost



Acknowledgements

BROADCOM®

UNIVERSITY OF CAMBRIDGE

LEVERHULME TRUST

Isaac Newton Trust