FIB Scaling: A Switch/Router Vendor Perspective

Greg Hankins

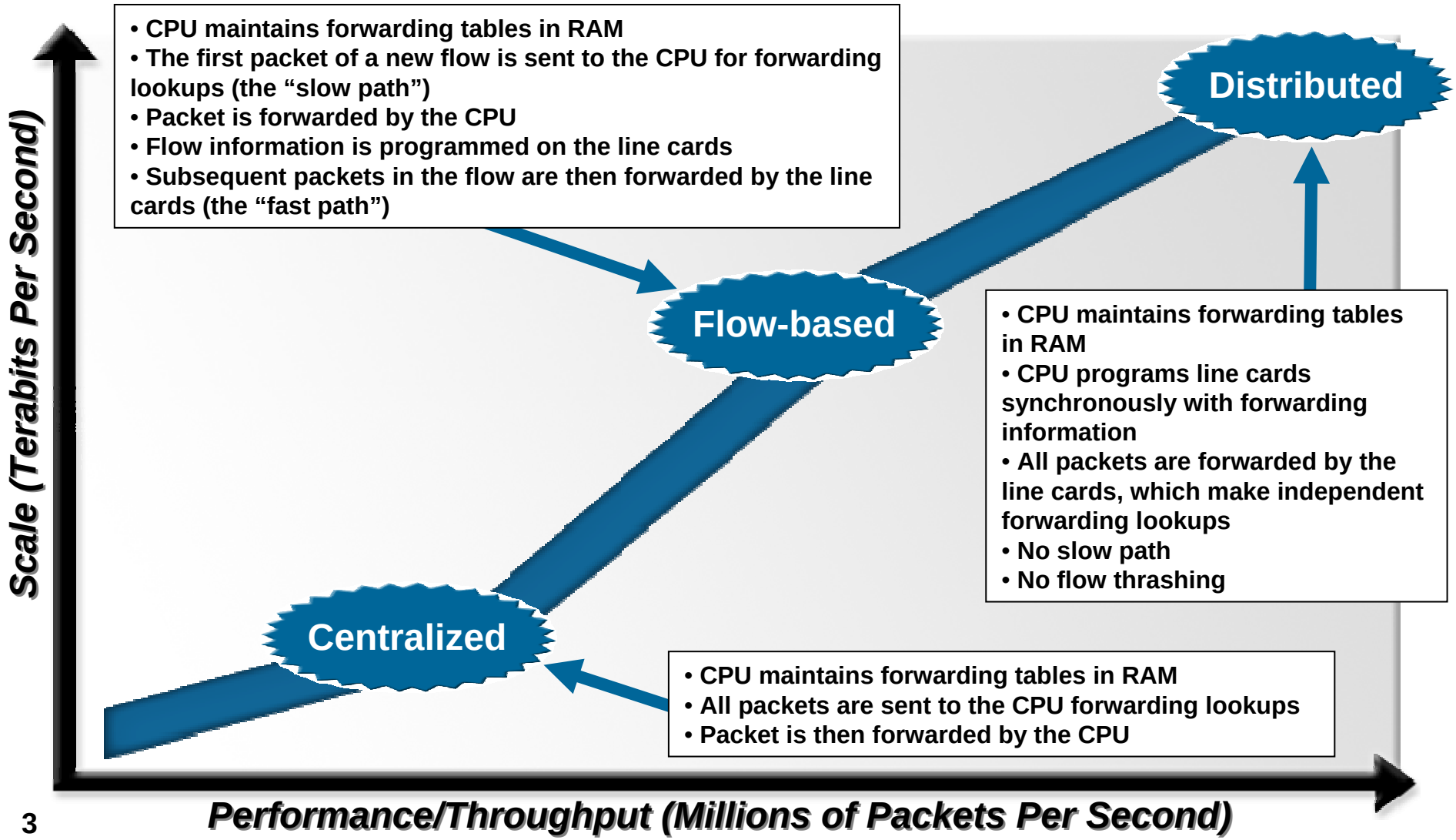
<ghankins@force10networks.com>

FIB Limits BOF

What Issues Are We Facing?

- Operators:
  - FIB scaling
  - FIB churn
  - Costs
- Vendors:
  - System design constraints and tradeoffs
  - Component technology
  - Costs

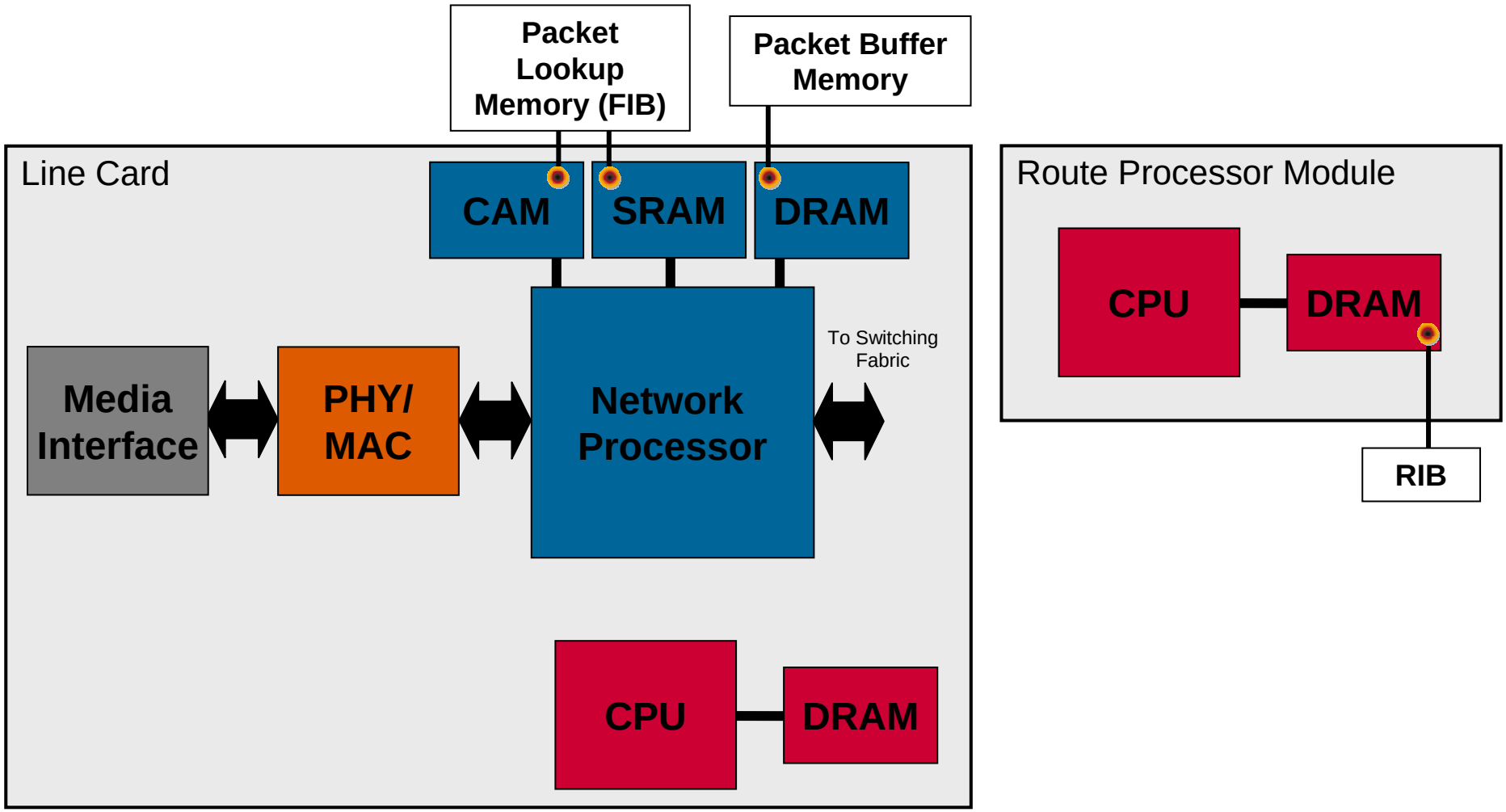
Evolution of Forwarding Architectures



Lookup and Forwarding Memory Overview

	TCAM	SRAM	DRAM
Type	Ternary Content Addressable Memory	Static Random Access Memory	Dynamic Random Access Memory (Must be Refreshed)
Primary Function	Search	Storage	Storage
Speed	7ns Search Time	3ns Access Time	50ns Read Latency 3ns Access Time
Cost per Bit	8 x – 10 x > SRAM		4 x < SRAM
Power Consumption	2 x – 3 x > SRAM		1/4 x < SRAM

Types of Lookup and Forwarding Memory Used in a Switch/Router



What Needs to be Stored to Forward Packets?

- Need to store all the stuff needed for packet lookups in hardware to forward at line rate with all features enabled
  - 10 GbE: packet every 67ns
  - 100 GbE: packet every 6.72ns
- This isn't just the FIB, need to store
  - MAC table
  - IPv4 FIB (unicast and multicast, VPN)
  - IPv6 FIB (unicast and multicast, VPN)
  - MPLS labels (including VC labels)
  - ACLs (ingress and egress)
  - QoS policies
  - System flows

What Needs to be Stored to Forward Packets?

- Also need to update the hardware forwarding information as fast as possible
  - Avoid blackholing traffic
  - Avoid writing a hardware entry when the FIB just changed again in software
- There is a lot of housekeeping that needs to take place to update an entry
  - Updates from protocols into RIB
  - Route selection into FIB
  - Next hop ARP resolution
  - Send update to line cards
  - Write to hardware

This takes ms

This takes µs

- Fast CPUs and distributed processing gives us update times on the order of ms

CAM (Search)

- CAM means TCAM in the networking industry
- Versatile data storage, no need to worry about how you write the data
  - Stores data and mask arrays for wild card searches
- Always get back an answer in the same search time, regardless of the complexity of the question
- Search returns the entry number if the data is found, which is then looked up in SRAM
- Using current 18Mbit CAM technology you can store
  - 512K IPv4 entries or
  - 128K IPv6 entries

207.69.188.185

CAM

Entry	Prefix	Mask
0	128.61.0.0	16
1	130.207.0.0	16
2	**207.69.0.0**	**16**
3	209.86.0.0	16

SRAM and DRAM (Storage)

- SRAM
  - Stores forwarding information (next hop, egress port, filtering tables, counters, linked list of packets in buffer)
  - Uses result from CAM lookup to find forwarding information used in the internal packet header
- DRAM
  - Useful when you need to read a burst of data
  - Need to optimize data layout in order to minimize read latency
  - Used for storing the RIB in memory on the CPU and for packet buffering on line cards

207.69.188.185

CAM

Entry	Prefix	Mask
0	128.61.0.0	16
1	130.207.0.0	16
2	**207.69.0.0**	**16**
3	209.86.0.0	16

SRAM

Index	Next Hop	Egress Port	Other Stuff
0	00:0C:6E:59:47:2B	te0/0	…
1	00:50:56:C0:00:01	te1/0	…
2	**00:0C:29:14:5E:83**	**te0/3**	**…**
3	00:50:56:C0:00:08	te0/1	…

What Can We Do Today in Hardware?

- Support flexible CAM allocation schemes
  - Typically use more L2, or more L3 addresses in a network and don't need huge tables for both
  - CAM partitioning schemes can be implemented in software to give operators choices on how the memory is allocated in hardware
- Use multiple CAMs
  - Increases complexity, power, cost, and takes up lots of board space
  - Using current technology, CAMs can scale to 2M IPv4 or 512K IPv6 prefixes
- Use SRAM instead of CAMs
  - Need to implement lookup mechanisms separately
  - Use data structures (hash tables or trees) in SRAM, but there are always data structure issues
  - Hard to implement ACLs in SRAM, CAMs are especially well suited for ACL lookups because they use masks

What Can We Do Today in Software?

- Intelligent population schemes can be written in software for prefixes in the FIB with the same next hop
- The FIB only cares about the next hop
- Install one entry for a prefix with a longer mask
  - A /20 and a /23 have the same next hop, only install the /20 in the FIB
- Aggregation of adjacent prefixes
  - Aggregate two /24s into a /23 in the FIB
- Increases complexity significantly

Technology Coming Soon

- 130nm processes have been widely available, 90nm now becoming widely available now for production
  - 2 x density
  - 30% less power consumption
  - 20% more performance
  - 65nm and 45nm expected to be widely available in the next few years
- CAM
  - 18Mbit today, higher density expected
  - Uses ASIC technology which depends on process size
  - Need faster lookups in the future, want to get back an answer in one clock cycle

Technology Coming Soon

- SRAM
  - 72Mbit today, 144Mbit expected
- DRAM
  - 288Mbit today, 576Mbit expected
- Faster interfaces that use less pins
  - Parallel chip-to-chip interfaces today (DDR/GDDR3)
  - SERDES interfaces already starting to ship (XDR/Rambus)
- Gaming consoles and high end computing are driving RAM technology
  - CPUs need fast SRAM for L2 caches
  - XDR and GDDR3 used on the PlayStation 3

Summary

- Hardware is designed to keep customers several years ahead of FIB memory exhaustion to avoid constant hardware upgrades
- CAMs are a universal search engine that give us maximum flexibility and constant lookup times today
- Many design aspects to consider, there are always tradeoffs
  - Cost
  - Power and heat
  - Board real estate space
  - Technology capabilities and complexity
- Number of entries today a cost and stability/complexity problem, not so much a technology problem
- Higher capacity and faster components are coming that will help us scale in the near term
- In the long term FIB explosion is a concern, and we'll need to look at the way we do things

Thank You