Packet Scheduling for Deep Packet Inspection on Multi-Core Architectures

Terry Nelms and ***Mustaque Ahamad***
Georgia Institute of Technology
ANCS'10

Nslab Seminar, Apr. 10th, 2013
Presenter: Zhi Liu

Outline:

- Introduction
- Background
- DPI packet scheduling algorithms
- Performance evaluation
- Conclusion

Introduction:

- DPI application
- PAM, software library using DPI to detect malicious network traffic
- PAM performance scale up
  - More CPU cycles
  - Thread level parallelism(multi-core, hardware threads)
- Packet scheduling problem among multi-thread DPI
  - TCP transition must be processed serially
  - Performance
- Contributions
  - Designed and implemented 2 new DPI scheduling algorithms
  - Algorithms evaluation
  - Scheduler design principles

Background:

- Packet Scheduling Overview
  - Packet based scheduling
    - E.g. round-robin
    - Packets evenly distributed but cache inefficient
  - Flow based scheduling
    - Maintains a table of active flows, new flows assigned based on load
    - Per-flow packet order maintained
    - Fixed flow-thread affinity, flow table maybe large

Background:

- Packet Scheduling Overview
  - Fixed hash scheduling
    - Direct hash and indirect hash based on 5-tuples
  - Adaptive hash scheduling
    - E.g. Receive-side scaling(RSS)
    - Utilizes indirect hash and re-calculate new indirect table when imbalance is detected
  - Flow burst scheduling
    - Maintaining a flow table that only contains the flows(currently in the system) and their processing threads
    - Flows are not fixed to a thread, e.g. between packet bursts
    - When a packet arrives:
    - If the flow exists in the table, assign the packet to the corresponding thread
    - If not, assign the packet to the least loaded thread

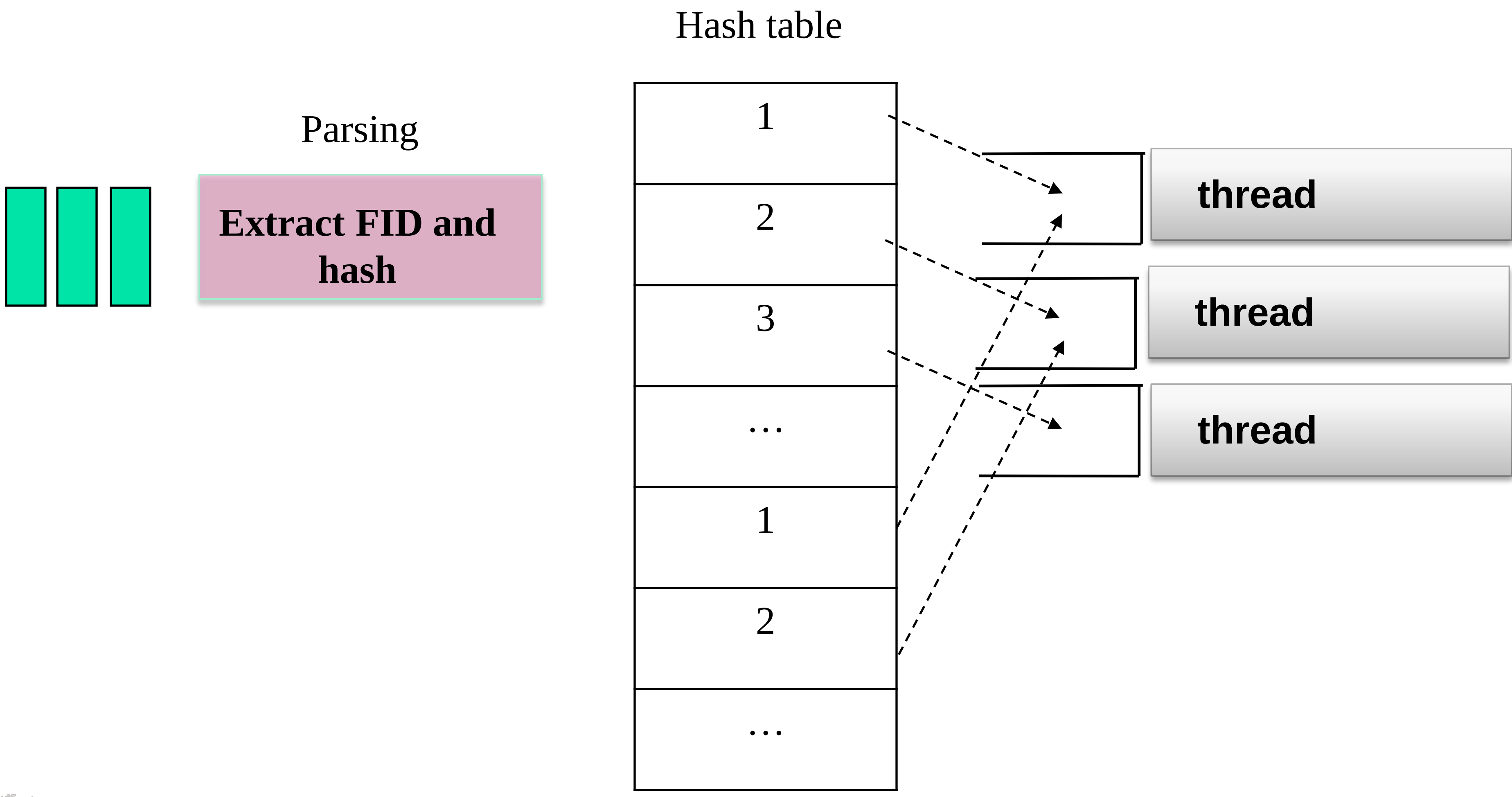
DPI Packet Scheduler:

- Packet Scheduling for DPI
  - The goals of DPI scheduler:
  - Maximize throughput, minimize average latency, bound maximum latency, minimize reordered packet number within a flow

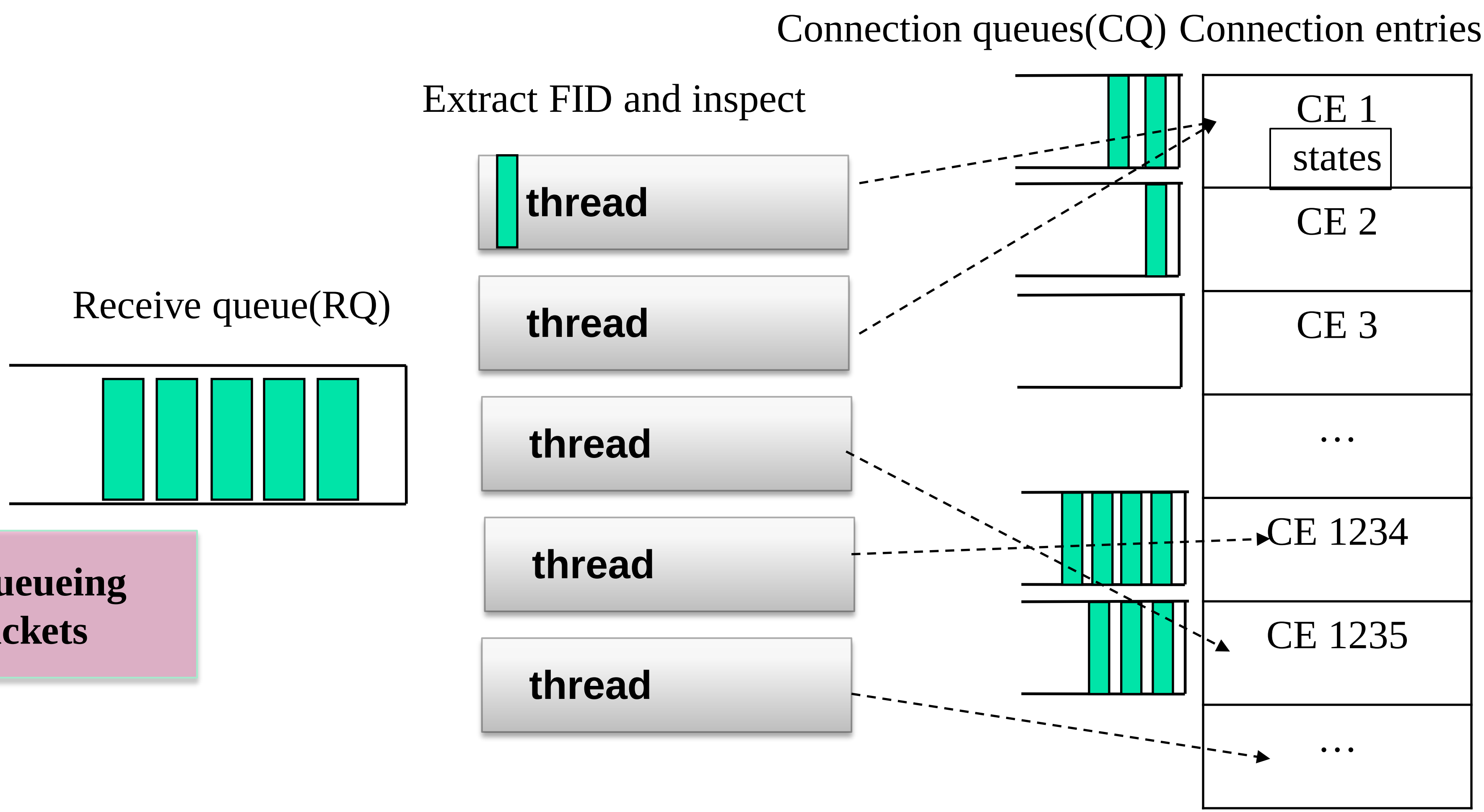
DPI Packet Scheduler:

- Ideal packet scheduler properties
  - Load balancing: work evenly distributed across all threads
  - Low scheduling overhead
  - Per-flow ordering
  - Cache affinity
  - Minimal packet delay variation
- Evaluating three scheduling algorithms
  - Direct hash
  - Packet handoff
  - Last flow bundle

Algorithm: Direct Hash(DH)



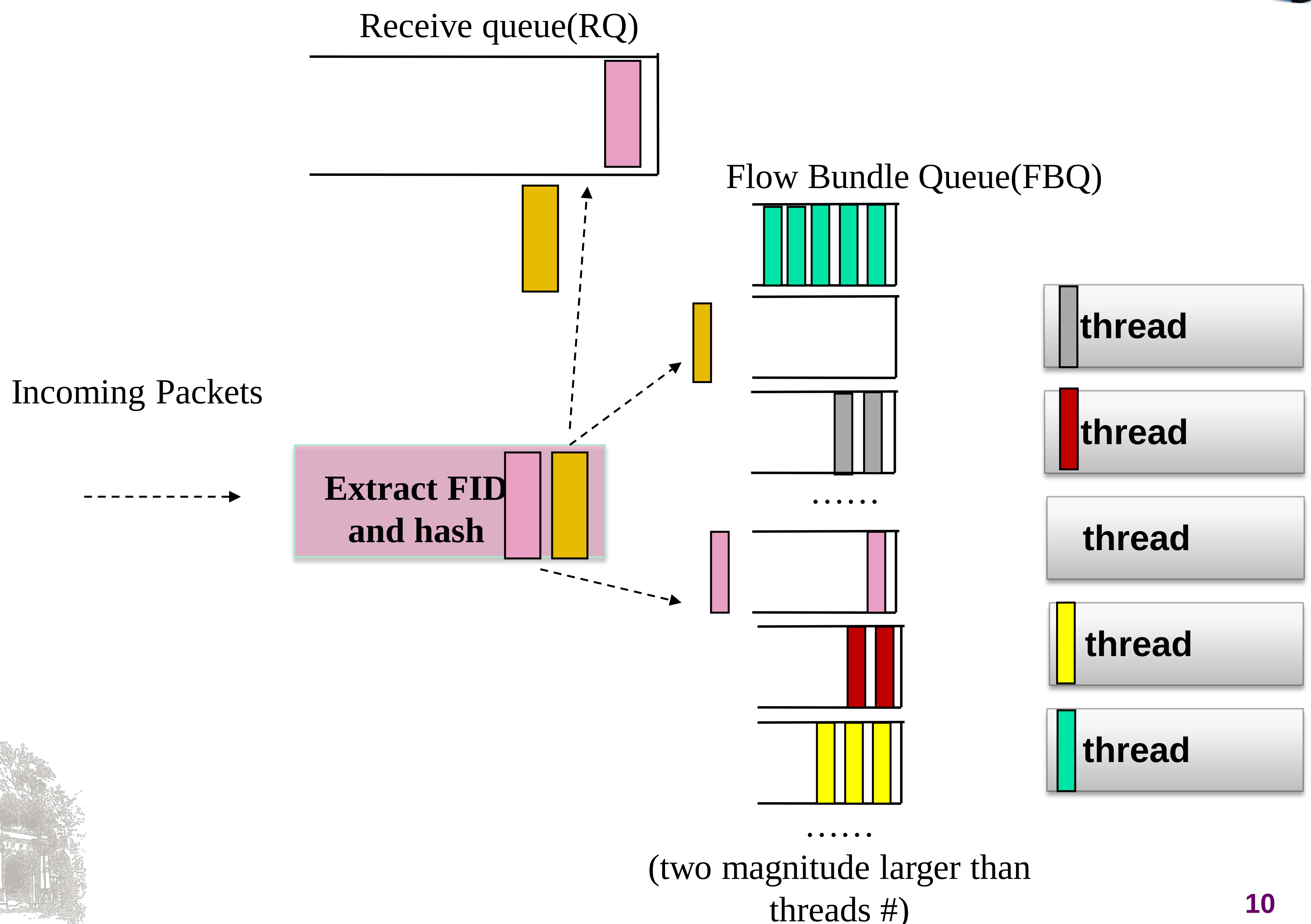
Algorithm: Packet Handoff(PH)



Enqueueing packets

One flow per connection entry

Algorithm: Last Flow Bundle(LFB)



Performance Evaluation:

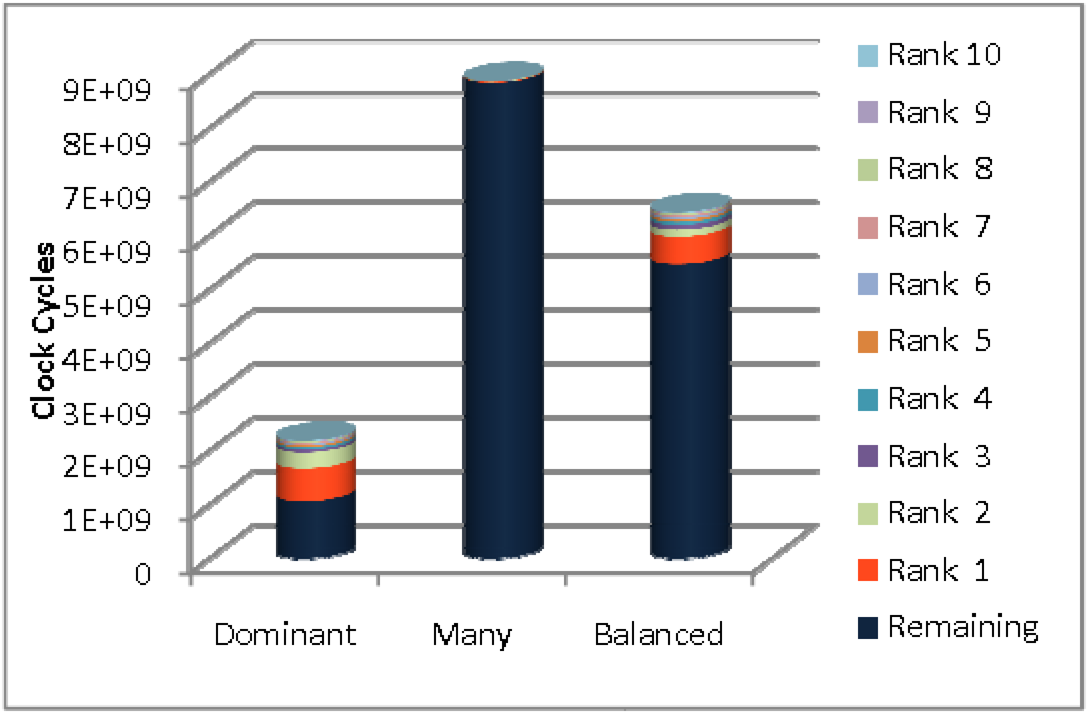
- Throughput evaluation
  - Maximum raw throughput
    - How it is calculated?
  - Maximum scaled throughput
- Latency evaluation
  - Average packet latency
  - Maximum packet latency
- Evaluation Platform
  - Two 2.53GHz Intel Quad-Core Xeon E5440 processors, 4G RAM
  - Each core has its own L1(32KB I, 32KB D),L2(256KB)
  - 8cores share 8MB L3 cache

Performance Evaluation:

- Network Captures
  - Real traffic captured from the location where DPI appliance is deployed
  - Three traffic captures:
  - Dominant
    - Dominated by one TCP connection(over 50% of pkts and 27% of clocks)
  - Many
    - Large amounts of small flows and spread throughout the capture
  - Balanced

Table 1: Network Capture Attributes

Caps	Packets	Conns.	Avg. Mb/s	Max. Mb/s	Avg. CPP	CD
Dominant	454,988	2,224	13	38	4,814	7
Many	1,567,397	105,691	27	74	5,658	50
Balanced	1,236,710	43,357	57	75	5,203	29



Evaluation: Throughput

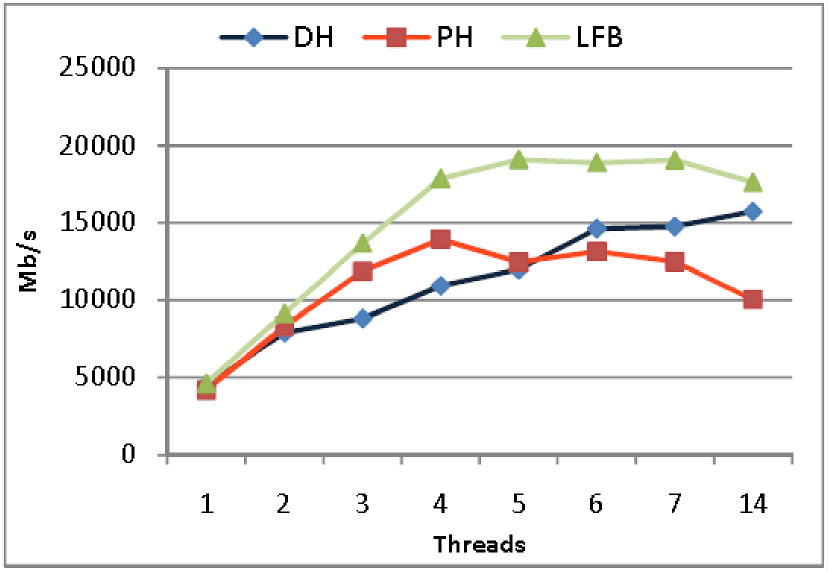


Figure 2: Dominant Capture Raw Throughput

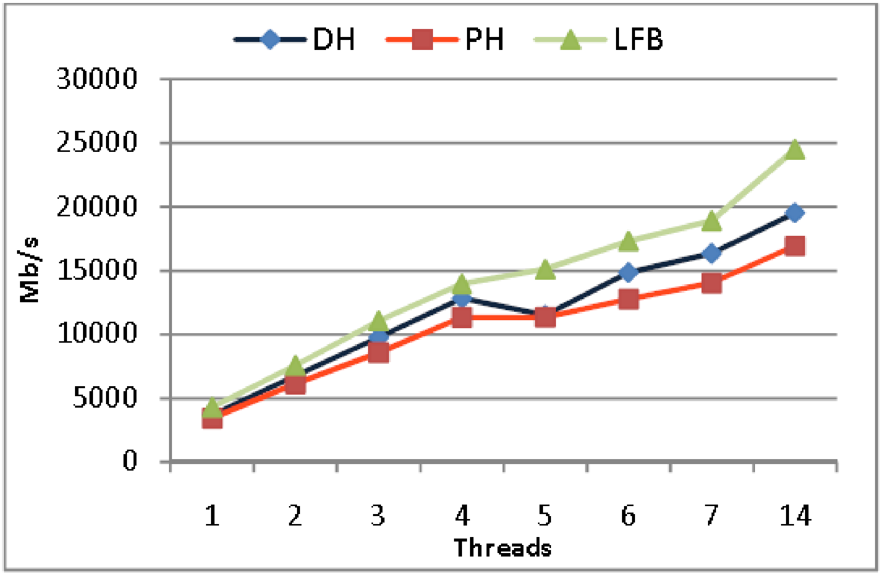


Figure 3: Many Capture Raw Throughput

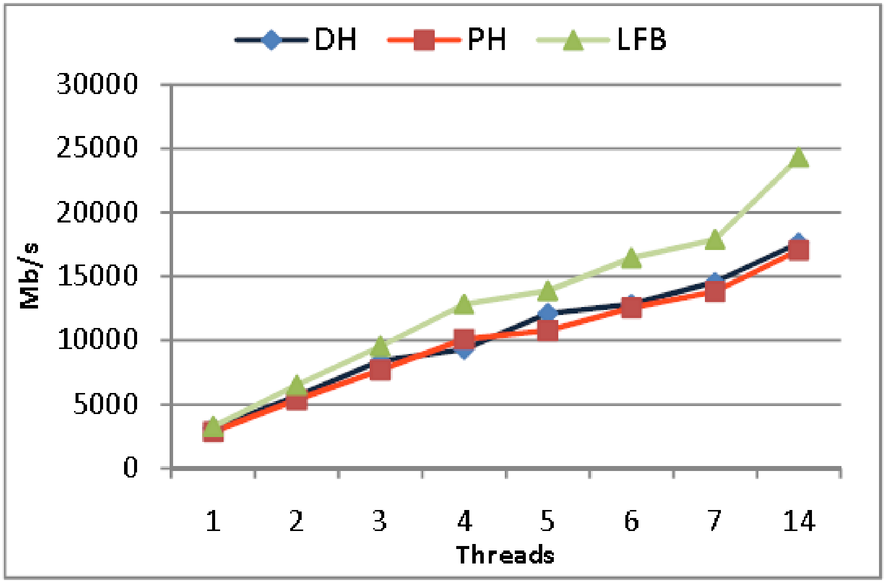


Figure 4: Balanced Capture Raw Throughput

Evaluation: Latency

- DH has the highest average latency for all three captures due to imbalance.
- LFB has the highest maximum latency for the Dominant and Balanced captures
  - During a burst of traffic, packets in the RQ packets on an index in the FBQ are processed.
  - This does not happen with the Many capture because of the high connection density.

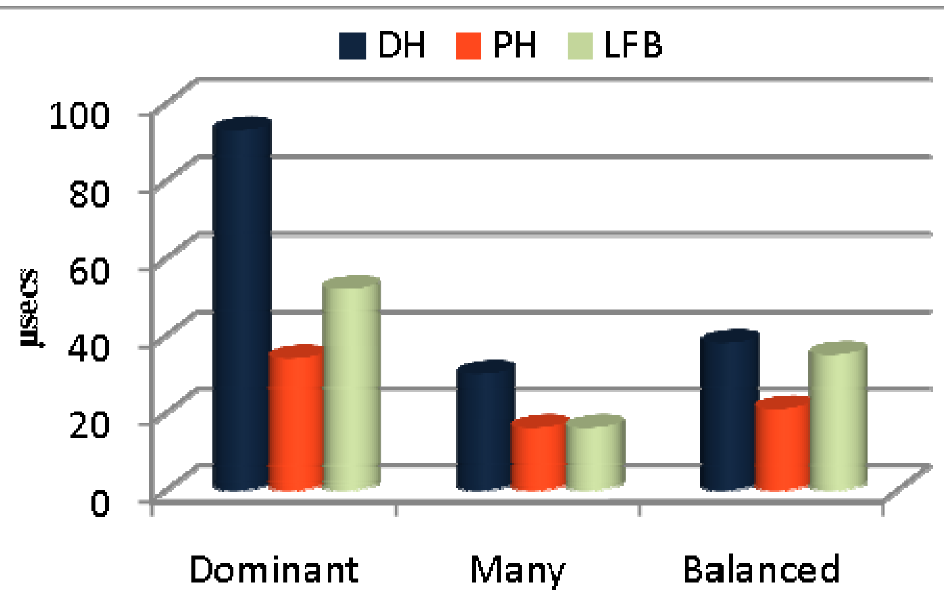


Figure 6: Average Latency

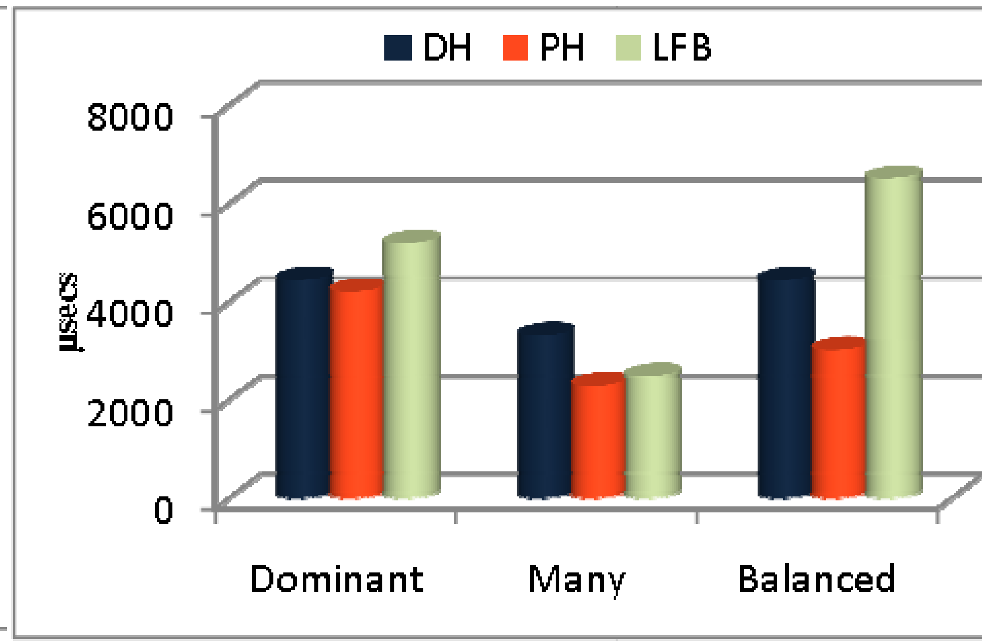


Figure 7: Maximum Latency

Evaluation: Cache Measurements

- For the Dominant capture, considerable cache misses for PH resulting from the large number of packet handoffs.
- For the Dominant capture ,DH and LFB have nearly identical cache miss.
- The Many capture produced the highest number of cache misses per packet for DH and LFB

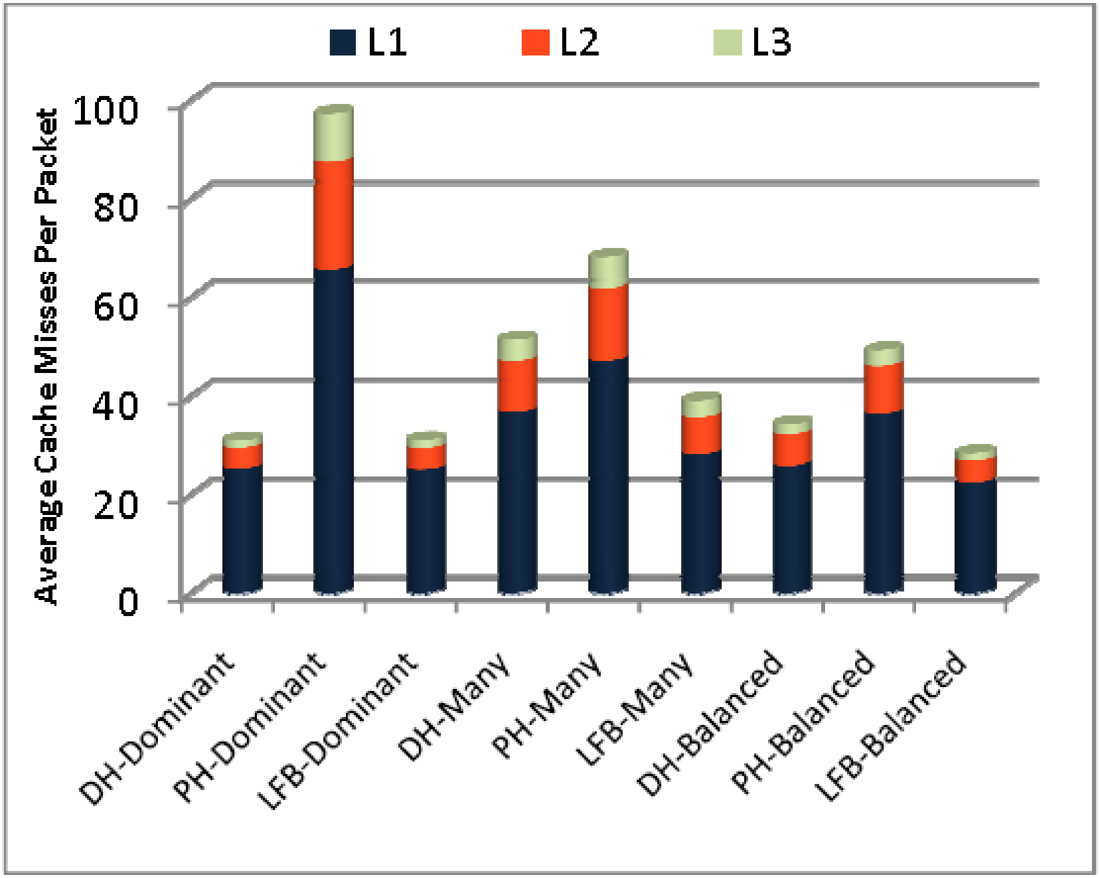


Figure 8: Average Cache Misses Per Packet

Evaluation: Cache Measurements

- Evaluate the throughputs with half of the L3 cache available.
- For the Dominant capture, LFB performance only declined by around 7%.
  - Because the state for the dominant connection easily fits in the smaller L3 cache and it is accessed enough to keep it in cache.
  - As for DH, it experienced a higher percentage decrease.
- The results for the Many capture show throughput decrease for all three packet scheduling algorithms.

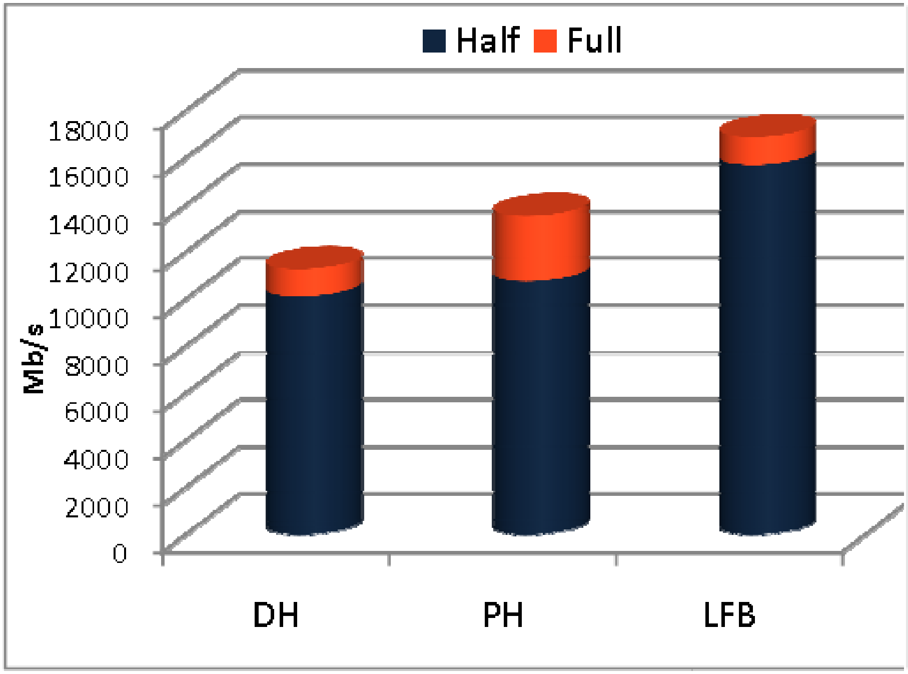


Figure 10: Dominant Capture with ½ L3 Cache

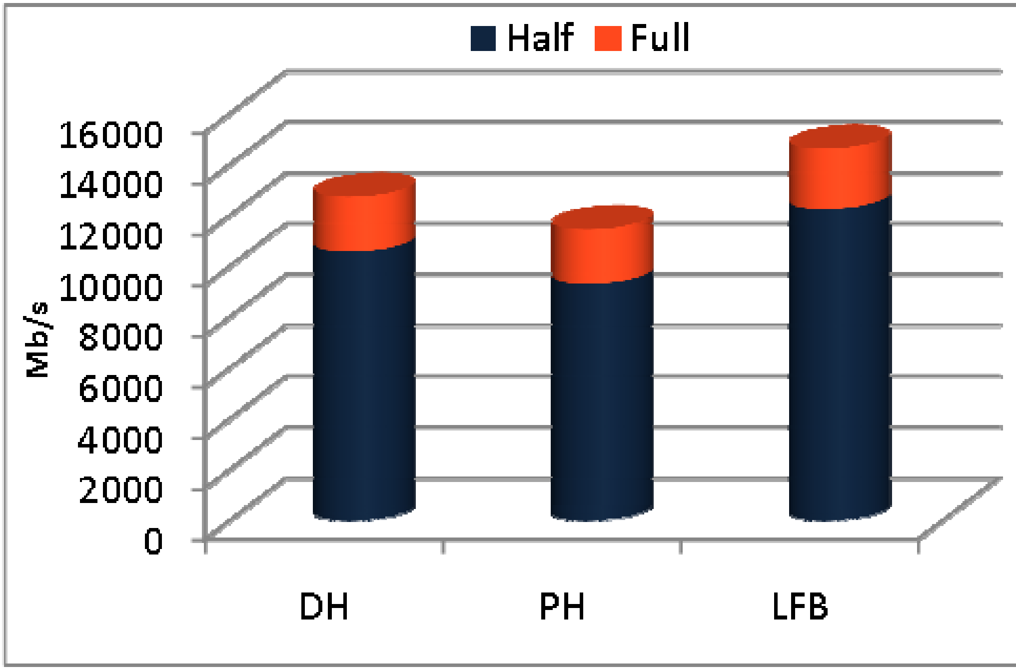


Figure 11: Many Capture with ½ L3 Cache

Conclusion:

- Design and implement two packet scheduling algorithms, each maximizes a different attribute of our ideal scheduler
- LFB maximizes cache affinity, outperforms the other two in terms of throughput for all network captures
- Results show the importance of cache affinity in packet scheduling

Thanks & Questions