Revisiting router architectures with Zipf

Steve Uhlig
Queen Mary, University of London, UK

Nadi Sarrar, Anja Feldmann
Telekom Innovation Laboratories/TU Berlin

Rob Sherwood, Xin Huang
Deutsche Telekom Labs USA
Agenda

• Motivation
  • Revisiting IP routers architecture
  • Opportunities and challenges
• Evaluation
• Conclusion
Software-defined networking

- Beyond today’s monolithic network equipment
- Separation of control and data plane through software modularity, e.g., Linux
- Do not change existing control plane
- Principles
  - Communication channel between forwarding engine and remote controller
  - Expose network equipment capabilities, e.g., TCAM, QoS
Agenda

- Motivation
- **Revisiting IP routers architecture**
- Opportunities and challenges
- Evaluation
- Conclusion
• System combines (1) fast switching hardware with (2) software router:
  - Fast switch handles most of the traffic with a few entries, i.e., fast path.
  - Software handles control plane and remaining traffic, i.e., slow path.

• Our approach: take advantage of the traffic properties.
Agenda

• Motivation
• Revisiting IP routers architecture
• Opportunities and challenges
• Evaluation
• Conclusion
Opportunity: Zipf

- Data
  - Transcontinental link: 150Mbps link (MAWI), 3.5 days
  - Residential ISP: 1Gbps link, 2 days
  - IXP: > 1Tbps, 4 days

- Observation: Most traffic captured by limited number of destination prefixes

⇒ Opportunity: Existing switching hardware can do it
Slow path: challenge?

- Assume knowledge of the future traffic
- Slow path rate as a function of number of heavy-hitters
- A few thousand heavy hitters enough to keep slow path rate low
- Limited variations across traces

⇒ With a few thousand flows, slow path rate can be kept low
Challenge: churn

• Assumption: knowledge of the future traffic

• Question: what is the expected churn rate?

• Answer: proportional to number of heavy-hitters

⇒ Challenge: keep churn low and most traffic on the fast path
Agenda

- Motivation
- Revisiting IP routers architecture
- Opportunities and challenges
- Evaluation
- Conclusion
Caching and churn

- Traditional caching:
  - LRU: replace least recently used entry upon miss
  - LFU: replace least frequently used entry upon miss
  - Always replace entry upon miss
  - Optimizes for hit rate, not churn
  - High churn when small cache

➡ Guideline: do not react immediately upon misses
Traffic Offloading

• **Tame natural churn of heavy-hitters**

• **Traffic Offloading (TFO)**

  Algorithm:
  
  - Monitor traffic at multiple time-scales
  
  - Select heavy-hitters that are expected to lead to low churn
  
  - Trade-off offloading gain with churn
Churn

• Traditional caching
  - Always replace entry upon miss
  - Good when cache can keep most heavy-hitters
  - Leads to high churn for low number of heavy-hitters
• TFO keeps churn much lower than bin-optimal and caching
• Combination of caching and TFO is ideal
Slow path

- LFU shows importance of heavy-hitters dynamics over short time-scales
- LRU and TFO close to optimal
- Slow path rate low for a few thousand heavy-hitters
TFO: churn

- Churn depends on traffic aggregation
  - IXP: a few changes per second
  - ISP: 10’s of changes per second
  - Transcontinental link: up to 100 of changes per second

- TFO tames the churn

⇒ Feasible on today’s OpenFlow-enabled switches
TFO: slow path

- Load can be handled by commodity PC
- Could be done on better embedded switch CPU
- Scaling up
  - Routebricks
  - Packetshader
  - PEARL
  - Traditional router

⇒ Feasible on today’s commodity hardware
Agenda

• Motivation
• Revisiting IP routers architecture
• Opportunities and challenges
• Evaluation
• Conclusion
Conclusion

• **Revisiting router architecture through SDN**
  - Leverage traffic properties (Zipf)
  - Combine open-source routing with fast and cheap switching hardware

• **TFO algorithm**
  - Beyond traditional caching: carefully select the right heavy-hitters
  - Keep both churn and slow path rates low

• **Scalability: get the best out of your hardware!**
Future work

• Next generation line-cards and routers
  – TCAM-based: Will TCAM become cost and power-efficient?
  – Multi core-based: Will advances in virtualization provide performance and isolation?
  – Routing-forwarding interactions: how much churn on the data-plane is useful?

• Heavy-hitter selection
  • Traffic monitoring: scalable and flexible per-entry statistics
  • Flexible algorithms for improved churn