Technische Universität Berlin · An-Institut Deutsche Telekom Laboratories

# FIBIUM

## Towards Hardware Accelerated Software Routers

Nadi Sarrar · Anja Feldmann · Steve Uhlig · Rob Sherwood · Xin Huang

## Motivation

Can we build an inexpensive yet powerful router?

- Routers today are propietary, closed-source, and expensive devices.
- Both routers and switches provide packet switching at line rate.
- **OpenFlow** [1] provides a programmable data-path.
- **Quagga** [2] implements an open-source software router.

## Our Approach

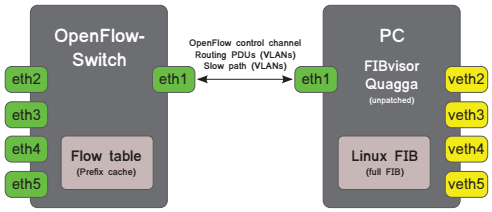
**FIBIUM** [3] is a novel architecture for designing routers in an alternative way.

- Internet traffic is heavy-tailed among prefixes – efficient **prefix caching** might be feasible with a sufficiently smart caching scheme.
- An **OpenFlow**-capable device switches traffic contributed by cached top prefixes. A PC handles the remainder.
- **Quagga** peers with other routers and maintains the full routing information base (RIB).
- **FIBvisor**, our OpenFlow controller, implements a glue layer to bring together the software router and the switch.

## FIBIUM Building Blocks

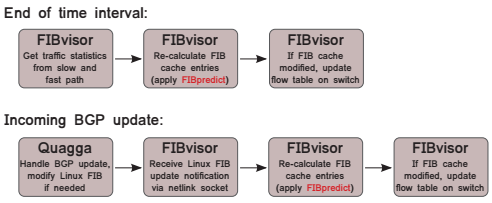
### Design

- **Port mirroring**: Trunked VLAN to create virtual siblings of physical router ports, allowing Quagga to operate.
- **Fast path**: Pure hardware-based switching of traffic towards popular prefixes.
- **Slow path**: Unpopular prefixes take the detour through the PC to let Linux decide on routing.




### FIBvisor

- **Prefix caching**: We propose **FIBpredict** as a caching strategy to keep a continuously updated set of popular prefixes not to overwhelm the slow path.
- **Traffic monitoring**: Keep traffic statistics for both, fast path and slow path.
- **FIB monitoring**: Monitor modifications on the Linux FIB by Quagga.




### FIBpredict

- **Select prefixes** for the fast path FIB (OpenFlow flow table) based on short, mid, and long-term traffic statistics to keep the adaptation time low while at the same time remain stable among the long-term top prefixes.
- **Reduce FIB churn** by dampening the fluctuation among less popular prefixes.
- **Update the fast path FIB** periodically, but react on BGP updates immediately.
- **Detect malicious activity** and consider countermeasures, e.g., discard traffic originated by a certain source using ACL's.

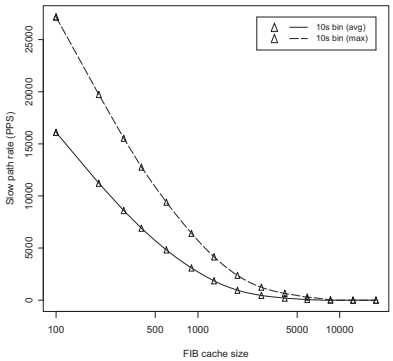
## Feasibility and Performance Evaluation

### Baseline experiments

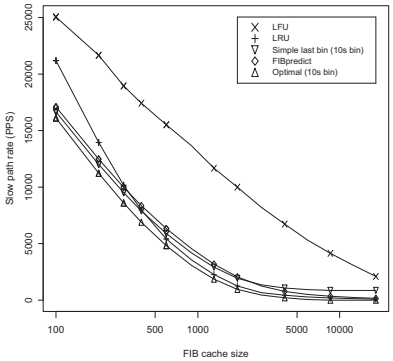
- Experiments based on 48hrs of anonymized packet-level data as seen by a customer aggregation router of a large European ISP.
- The optimal case with future knowledge supports our claim, that efficient prefix caching in theory is possible.

### Prototype evaluation

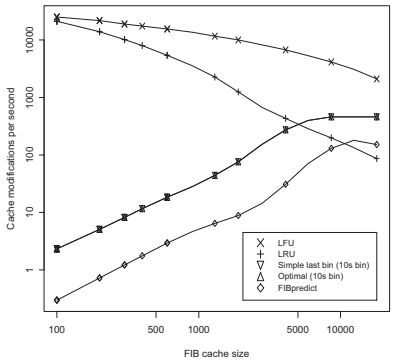
- Simulating FIBpredict results in a behavior which seems close to the optimal case.
- LRU and LFU, both reacting on individual cache misses, still show an idealized behavior, which is unfeasible in practice.




**Baseline slow path requirements.**




**Comparison of slow path requirements.**




**Churn for different caching schemes.**

## Future Work

- Evaluate the performance of FIBpredict with more data from different network locations.
- Measure the performance of different OpenFlow-capable devices, e.g., quantify the inherent penalty induced by flow table modifications.
- Deploy FIBvisor in an experimental facility with real users.

## References

[1] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. OpenFlow: Enabling Innovation in Campus Networks. In *ACM CCR* (2008).

[2] Quagga Routing Suite. `http://www.quagga.net`

[3] Feldmann, A., Uhlig, S., Sherwood, R., Sarrar, N. Route Caching System. Patent application 10007/TUB.