Hash-based Packet Selection in IP Networks

Synchronization of Hash-based Packet Selection in IP Networks

Tácio Santos
<tacio@zedat.fu-berlin.de>

Freie Universität Berlin, Computer Systems & Telematics
Prof. Dr.-Ing. habil. Jochen H. Schiller

Fraunhofer FOKUS Berlin, Network Research
Dr.-Ing. Tanja Zseby

INET Seminar
April 17, 2012

IP Networks

- ➩ Internet Protocol (version 4)

Hash-based Packet Selection

- ➩ network measurement technique

Synchronization

- ➩ periodical adjustments to the measurement system

WHY?

Some reasons are:

- Service Level Agreement validation
- Network management/engineering
- Security

➪ need to understand

ONE BIG CHALLENGE

- High data volume ➪ data reduction

HASH-BASED PACKET SELECTION

- Data reduction ➪ emulates random sampling
- Multi-point ➪ the same set of packets along its path
- Passive ➪ no packets are injected in the network
- Configurable sampling rate ➪ scale measurements up/down

Multi-hop Packet Tracking*

Objective

Observe a packet's path and its experienced transmission quality throughout the network.

Applications

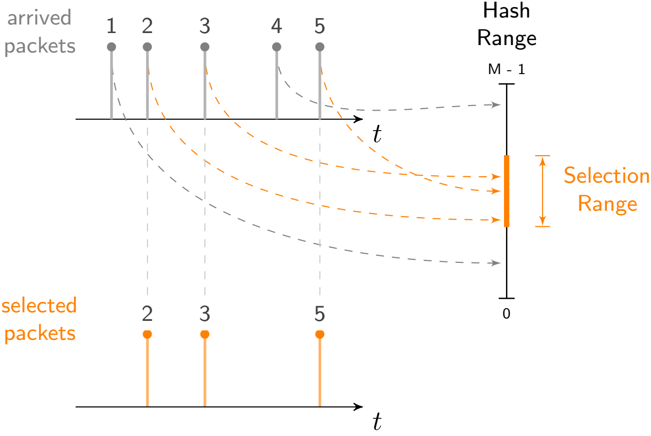
- Service Level Agreement validation
- Input to adaptive distributed algorithms
- Supervision of network experiments

Architectural Building Blocks

- Hash-based packet selection
- Time synchronization
- Standardized data export (IPFIX)
- Resource control with sampling rate adaptation

* http://www.fokus.fraunhofer.de/de/net/projekte/Packet_Tracking

Hash-based Packet Selection*

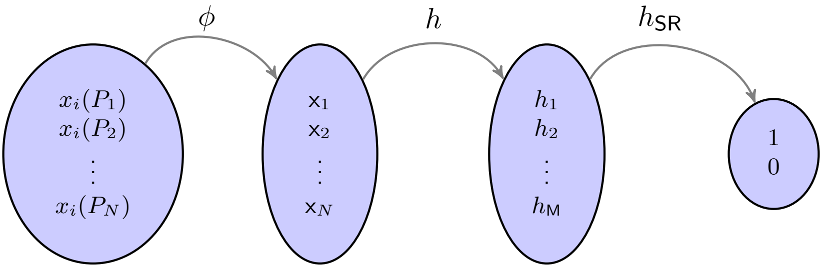


Sampling rate

$$r = \frac{|SR|}{M}$$

* N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation", IEEE/ACM TON, vol. 9, 2001

Hash-based Packet Selection*



Packets: $P_1, P_2, .., P_N$

Packet capture: x_i

Invariance function: ϕ

Hash function: h

Sampling function

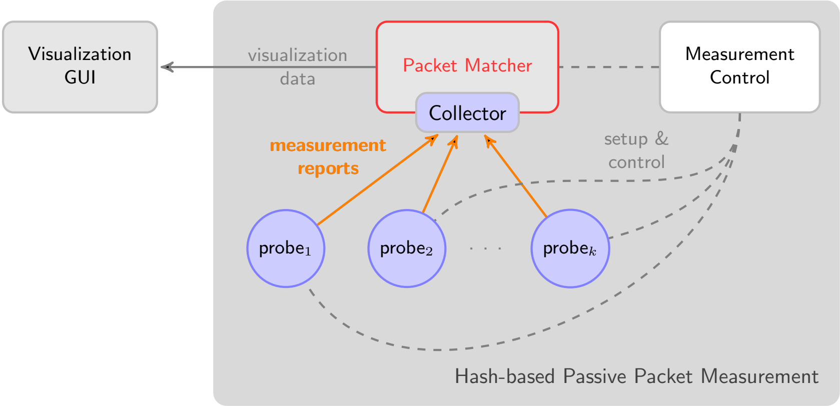
$$h_{SR} = \begin{cases} 1, & h_j \in SR \\ 0, & \text{otherwise} \end{cases}$$

Sampling rate

$$r = \frac{|SR|}{M}$$

* N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation", IEEE/ACM TON, vol. 9, 2001

FOKUS Packet Tracking System



Resources

- H: available bandwidth for the **measurement reports**
- O: processing capacity (reports per second **Packet Matcher** can process)

Sampling rate deviations

PROBLEM

Sometimes more packets are selected than expected. ⇨ system overload

CAUSES*

1. Inherently statistical deviations
2. Unanticipated behavior in the hash function
3. Packets deliberately crafted to be selected

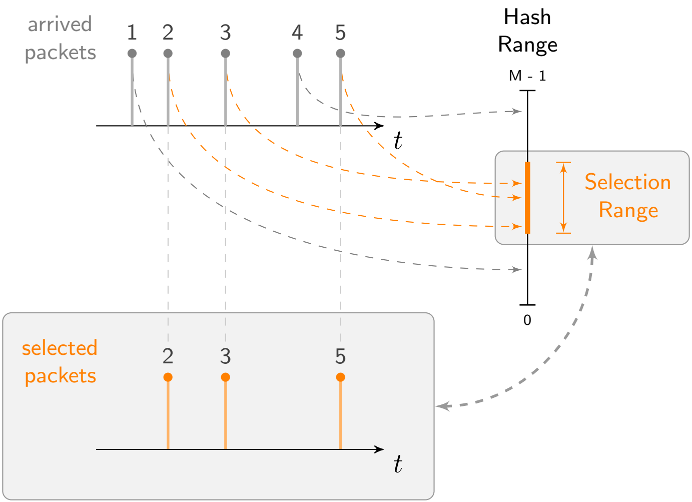
⇨ traffic dynamics

RESEARCH QUESTIONS

1. How to assure that the available bandwidth H is not exceeded due to the sampling rate deviations?
2. How to minimize these deviations considering that the system has a limited processing capacity O?

* T. Zseby, et al. ,"Sampling and Filtering Techniques for IP Packet Selection." RFC 5475 (Proposed Standard), 2009

Hash-based Packet Selection*



$r = \frac{|SR|}{M}$

* N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation", IEEE/ACM TON, vol. 9, 2001

Sample size synchronization

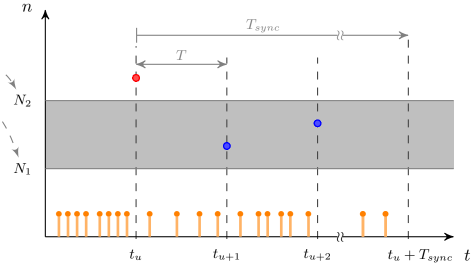
Assumption

Sampling rate deviations can be limited by periodic adjustments to the measurement system.

Approach

- sample size: $n(t)$
- dynamic boundaries: $n_{\text{sup}}(t), n_{\text{inf}}(t)$
- synchronization interval: T_{sync}
- number of selected packets:

$$\int_{t_u}^{t_u+T_{\text{sync}}} n(t)\,dt$$



$$\int_{t_u}^{t_u+T_{\text{sync}}} n_{\text{inf}}(t)\,dt \;\leq\; \int_{t_u}^{t_u+T_{\text{sync}}} n(t)\,dt \;\leq\; \int_{t_u}^{t_u+T_{\text{sync}}} n_{\text{sup}}(t)\,dt$$

Sample size synchronization

Sample size synchronization

Outcome

1. Can we keep the number of selected packets within predefined limits?

$$\int_{t_u}^{t_u+T_{\text{sync}}} n_{\text{inf}}(t)\,dt \leq \int_{t_u}^{t_u+T_{\text{sync}}} n(t)\,dt \leq \int_{t_u}^{t_u+T_{\text{sync}}} n_{\text{sup}}(t)\,dt$$

➪ YES, because of hash-based packet selection we can make $n(t) = 0$ when necessary

2. What happens if it takes too long until $n(t) = 0$?

➪ more packets are going to be selected

3. How to cope with this situation?

➪ model configuration changes

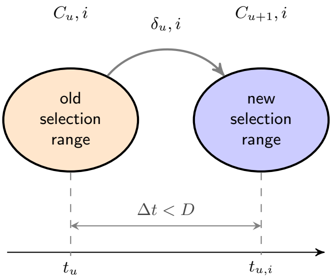
Configuration synchronization

ASSUMPTION

We cannot predict probe reaction time (IP, non RTOS, topology, traffic, ..)

APPROACH

Define a configuration delay limit D. Within D the measurement system is considered to be in a transient state.



OUTCOME

$$d_u = \max\left(t_{u,i} - t_u\right)_{i=1}^{k}$$
$$= \max\left(\pi\left(C_{u+1,i}\right) - t_u\right)_{i=1}^{k}$$
$$= \max\left(\pi\left(\delta_{u,i}(C_{u,i})\right) - t_u\right)_{i=1}^{k}$$

$C_{u,i}$: configuration

$\delta_{u,i}$: transition function

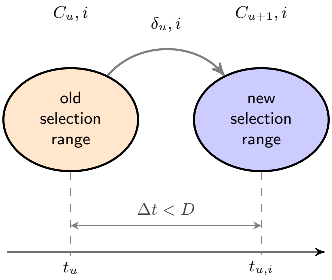
π : timestamp function

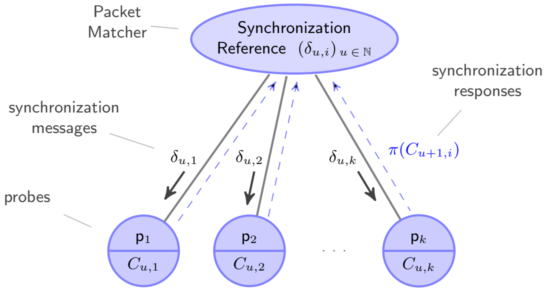
$$\forall u \in \mathbb{N}\ d_u < D$$

➪ D represents a global limit for the configuration delay

Configuration synchronization

OUTCOME





$$d_u = \max\Big(\pi\big(\delta_{u,i}(C_{u,i})\big) - t_u\Big)_{i=1}^{k}$$

$$\forall u \in \mathbb{N}\ d_u < D$$

Answers to Our Research Questions

1. How to assure that the available bandwidth H is not exceeded due to the sampling rate deviations?

   ➪ $$H = e \cdot \frac{\#\text{reports}}{\text{time interval}} = \frac{e}{T_{\text{sync}}} \sum_{i=1}^{k} \underbrace{\int_{t}^{t+T_{\text{sync}}} n_i(t)\,dt}_{\text{bounded}}$$

2. How to minimize these deviations considering that the system has a limited processing capacity O?

   ➪ Because the number of reports is bounded O is also bounded.

Implementation

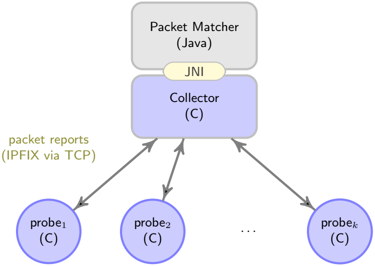
Previous Shortcomings

Probes

- static configuration
- missing detailed selection process and resource usage information

Matcher

- JNI based IPFIX Collector
- Can only receive measurement reports



My Contribution

Probes

- Redesigned to support asynchronous events and non-blocking I/O
  - ⇨ greatly simplified by libev*
- Use measurement report connections for synchronization
  - ⇨ no additional connections needed
  - ⇨ no problems with NAT

Matcher

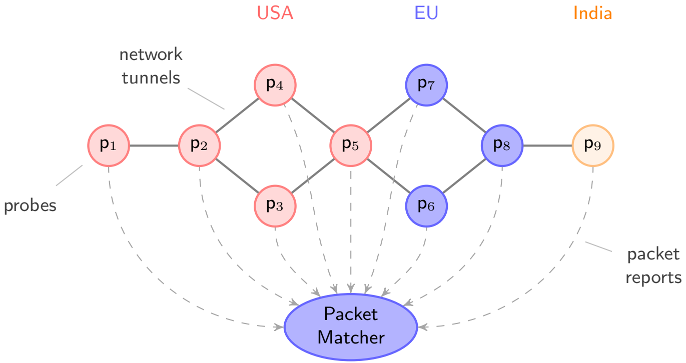
- IPFIX Collector in Java
  - ⇨ full control of connections
- Added probe management capabilities
  - ⇨ enabled run-time configuration of the measurement system

*M. Lehmann, "libev, a high performance full-featured event loop written in C"

Evaluation

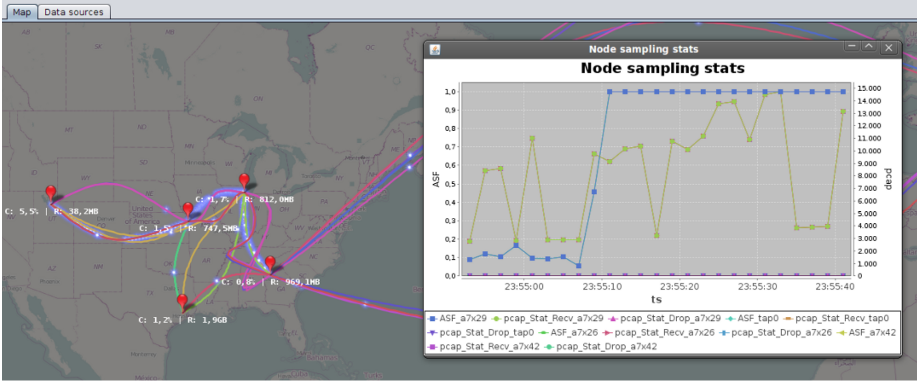
* SIGCOMM DEMO 2010, New Delhi, India

- Project: OneLab2
- Networks: PlanetLab, Vini, G-Lab
- Experiments: P2P and video streaming



* T. Santos, C. Henke, C. Schmoll, and T. Zseby, "Multi-Hop Packet Tracking for Experimental Facilities", SIGCOMM 2010

* SIGCOMM DEMO 2010, New Delhi, India



Packet Tracking Visualization

* T. Santos, C. Henke, C. Schmoll, and T. Zseby, "Multi-Hop Packet Tracking for Experimental Facilities", SIGCOMM 2010

Contributions & Outlook

Key Contributions

- An analysis of the impact of the sampling rate deviations on the FOKUS Packet Tracking System
- The formalization of the concepts of sample size synchronization and configuration synchronization applied to hash-based packet selection
- A configuration synchronization mechanism for IPFIX based measurement systems
- A proof-of-concept implementation of configuration synchronization for the FOKUS Packet Tracking System

Outlook

- Development and implementation of sample size synchronization algorithms
- Study of the impact of the configuration delay limit D on sample size synchronization

References

* FOKUS Fraunhofer Institute, "Multi-hop Packet Tracking", Website, http://www.fokus.fraunhofer.de/de/net/projekte/Packet_Tracking, retrieved on 2011-01-21.
* N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation", IEEE/ACM Transactions on Networking (TON), vol. 9, no. 3, p. 292, 2001.
* T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475 (Proposed Standard), Mar. 2009.
* M. Lehmann, "libev, a high performance full-featured event loop written in C", Website. http://software.schmorp.de/pkg/libev.html, retrieved on 2011-01-21 .
* T. Santos, C. Henke, C. Schmoll, and T. Zseby, "Multi-Hop Packet Tracking for Experimental Facilities", SIGCOMM 2010, 2010

Thank you for your attention!

Questions?