amsix
amsterdam internet exchange

# sFlow

Elisa Jasinska
elisa.jasinska@ams-ix.net

- Agenda
  - What is sFlow?
  - AMS-IX requirements
  - Existing software solutions
  - Performance issues
  - Software used at AMS-IX
  - Results

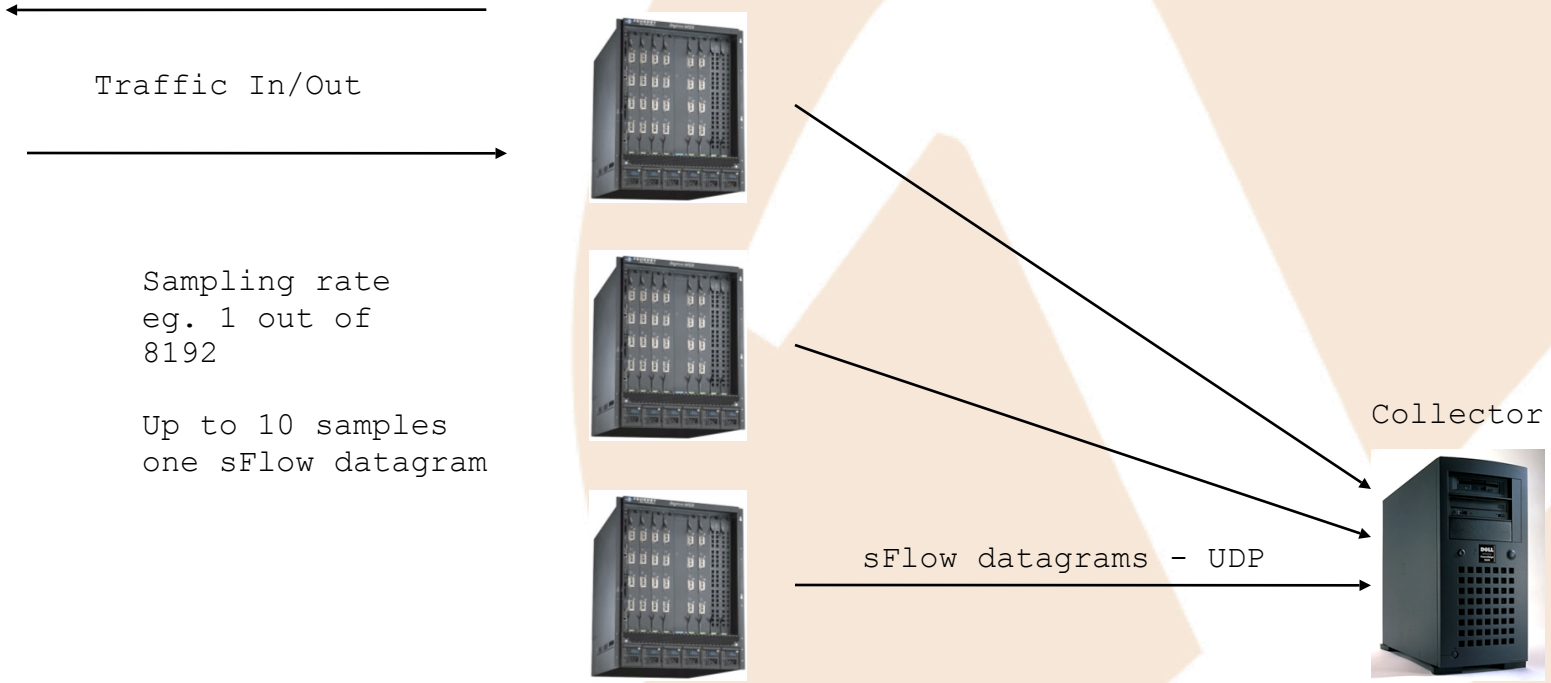
- What is sFlow?
  - Monitoring networks
  - Cisco IOS - NetFlow
  - Sampling mechanism, not “touching” every packet
  - Applicable to high speed networks (>= 1GE)

- What is sFlow?
  - sFlow datagrams sent via UDP
  - Datagram format standard defined in RFC 3176

- What is sFlow?
  - Flow samples
    - Whole captured packet (L2-L7)
  - Counter samples
    - Interface counters (octets/pkts/errors)

- What do we need?
  - Hardware supporting sFlow (eg. Switch)
  - Central server to collect the data
  - Software analyzing the received data

- How does it work?



- AMS-IX requirements
  - Use flow samples to:
    - Provide member-to-member traffic information
    - See growth of (or lack of) of IPv6
  - Due to high throughput a very efficient system is required

- Existing software solutions
  - Free software:
    - InMon – sflowtool
    - Pmacct
    - sFlow2MySQL
  - Commercial:
    - InMon – Traffic Sentinel

- Issues with existing software
  - Saving each sample to DB
  - No caching or preprocessing possible
  - Graphing with RRDtool overhead (same data saved twice)

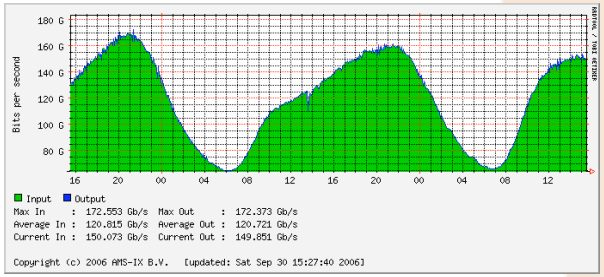
- Performance issues at AMS-IX
  - Traffic up to 180 Gb/s (30 Mpps)
  - ca. 3500 Samples per second
  - > each sample cannot be stored in DB

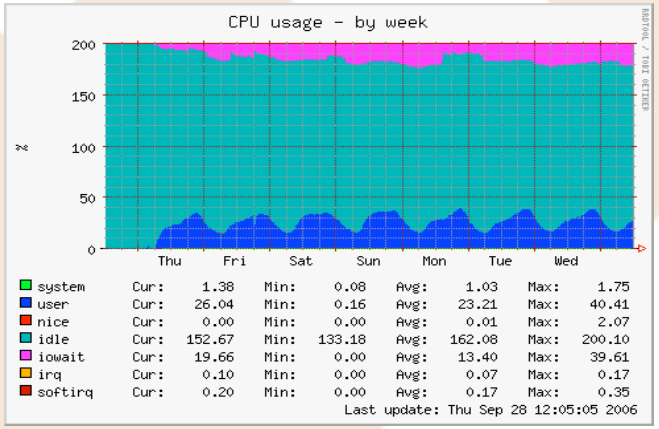
- Software developed at AMS-IX
  - Written in PERL
    - Easy to understand
    - Good integration with RRDtool
    - Due to PERL's re-use architecture (modules) lots of subtasks are already handled
    - Largest common denominator of a language understood at the AMS-IX NOC

- Net::sFlow
  - Decode sFlow datagrams
  - Supports sFlow version 2/4 and version 5
  - Single (exportable) function, decode()
  - Available on CPAN

- sFlow daemon
  - Based on module Net::sFlow
  - Receives UDP datagrams
  - Analyzes the information
  - Stores data to RRD files

- Performance results
  - CPU usage while decoding sFlow datagrams
  - Growing linearly with amount of packets / samples

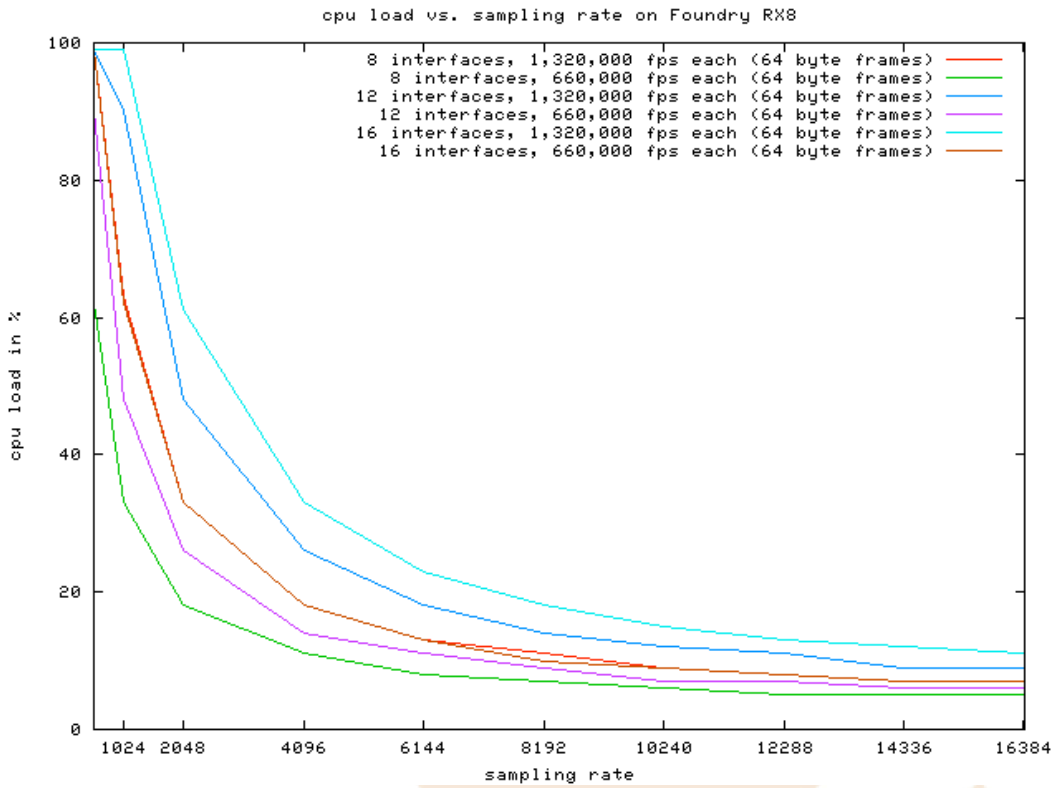


- Performance results
  - I/O performance while writing data
  - Currently:
    - Writing ca. 40 000 RRD files in 8 seconds
  - High load tests:
    - Writing 130 000 RRD files in 27 seconds
  - Max. at AMS-IX 160 000 conversations

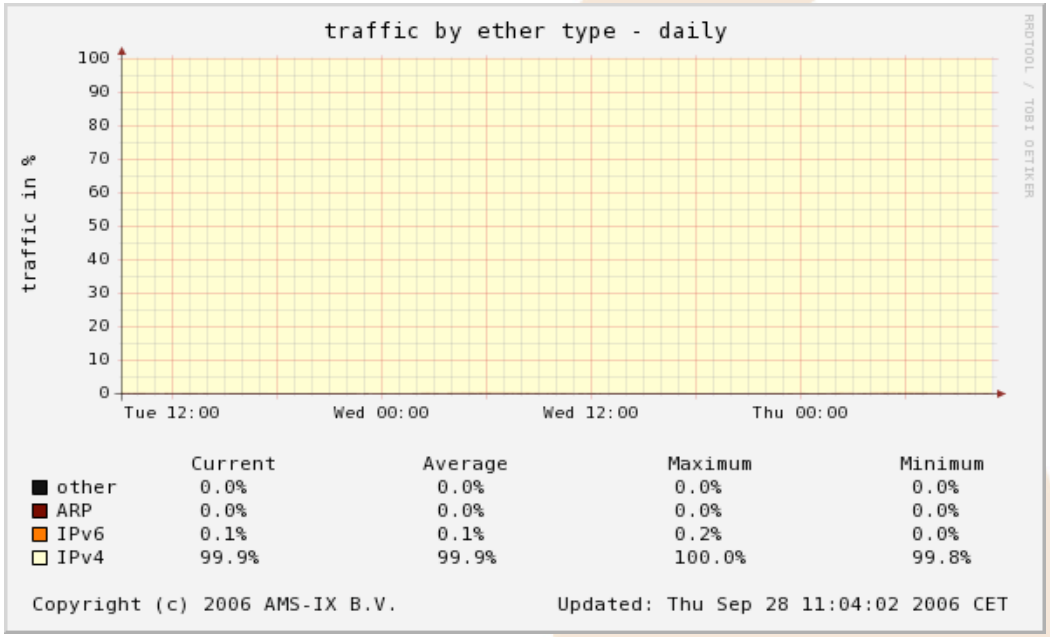
- Performance results - Foundry hardware
  - BigIron 15k
    - ASIC (Application-specific integrated circuit)
    - Switch CPU not affected
  - MG8 & RX*
    - Blade CPU affected

- Performance results - Foundry hardware

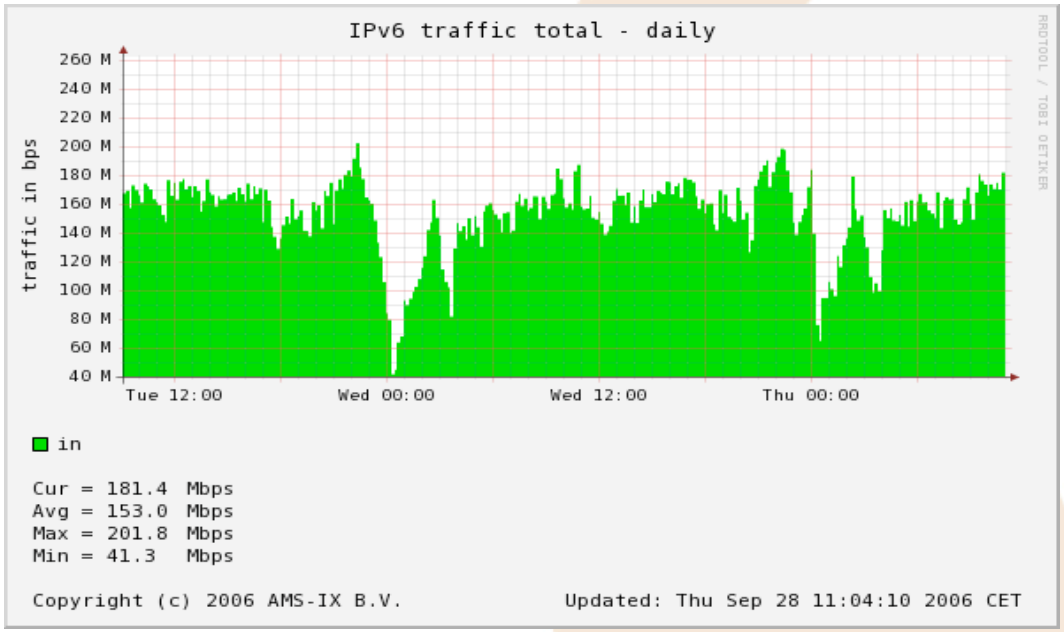


- Software developed at AMS-IX
  - Analysis:
    - Ether type graph – percentage of IPv4, IPv6, ARP and other
    - Total IPv6 traffic graph – in bps and pps
    - Member-2-Member analysis – in bps and pps

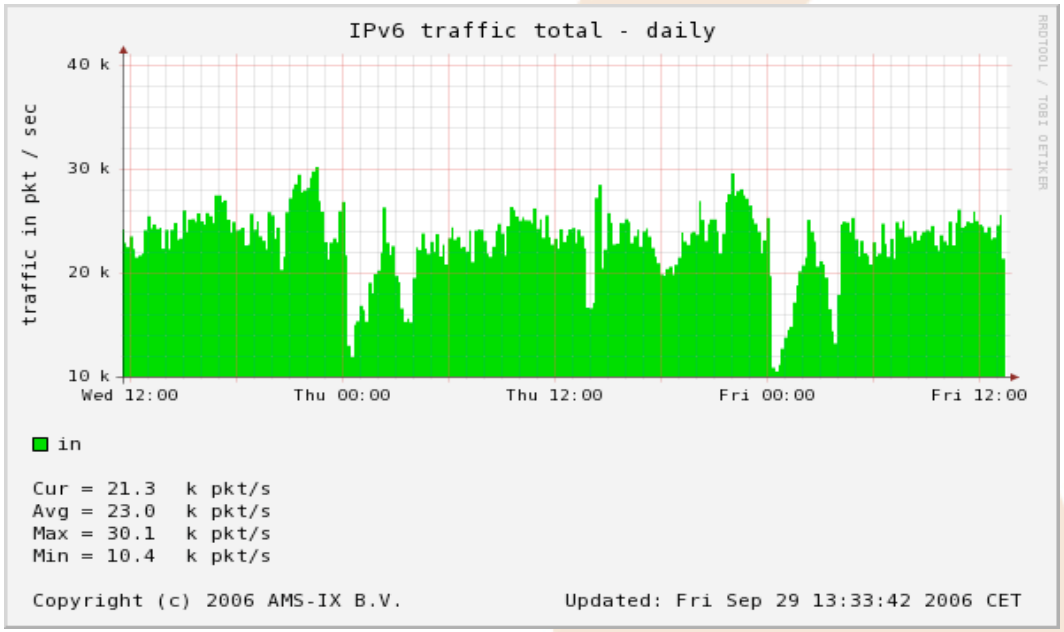
- Results – Ether type



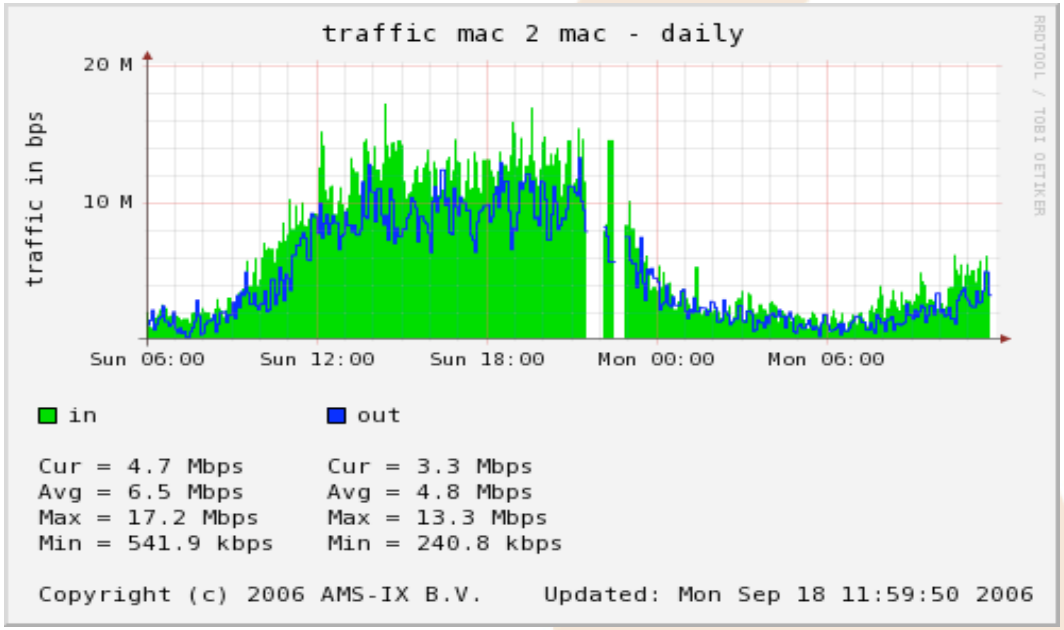
- Results – Total IPv6 traffic - bps



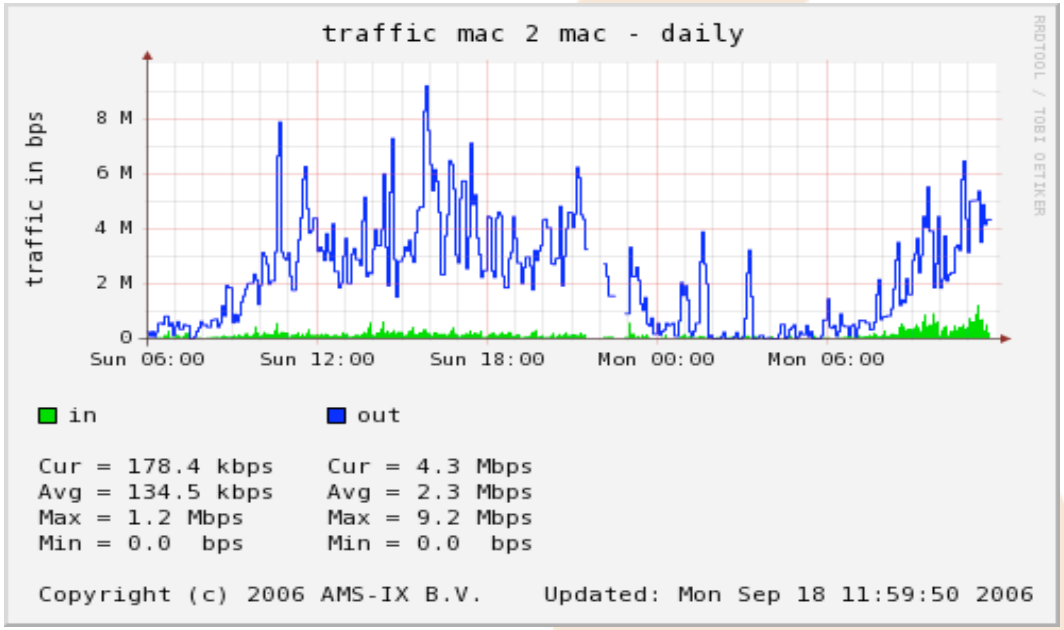
- Results – Total IPv6 traffic - pps



- Results – Member-2-Member traffic



- Results – Member-2-Member traffic



- Future plans
  - Use counter samples:
    - Separate interfaces
    - Aggregated links
    - Backbone links
    - Core network
    - ...
  - ...

# Questions ?

Comments / Requests / Ideas:

- elisa.jasinska@ams-ix.net
- sflow@ams-ix.net