sFlow – Why You Should Use It And Like It

NANOG 39

February 04-07, 2007

Richard A. Steenbergen
nLayer Communications, Inc.
<ras@nlayer.net>

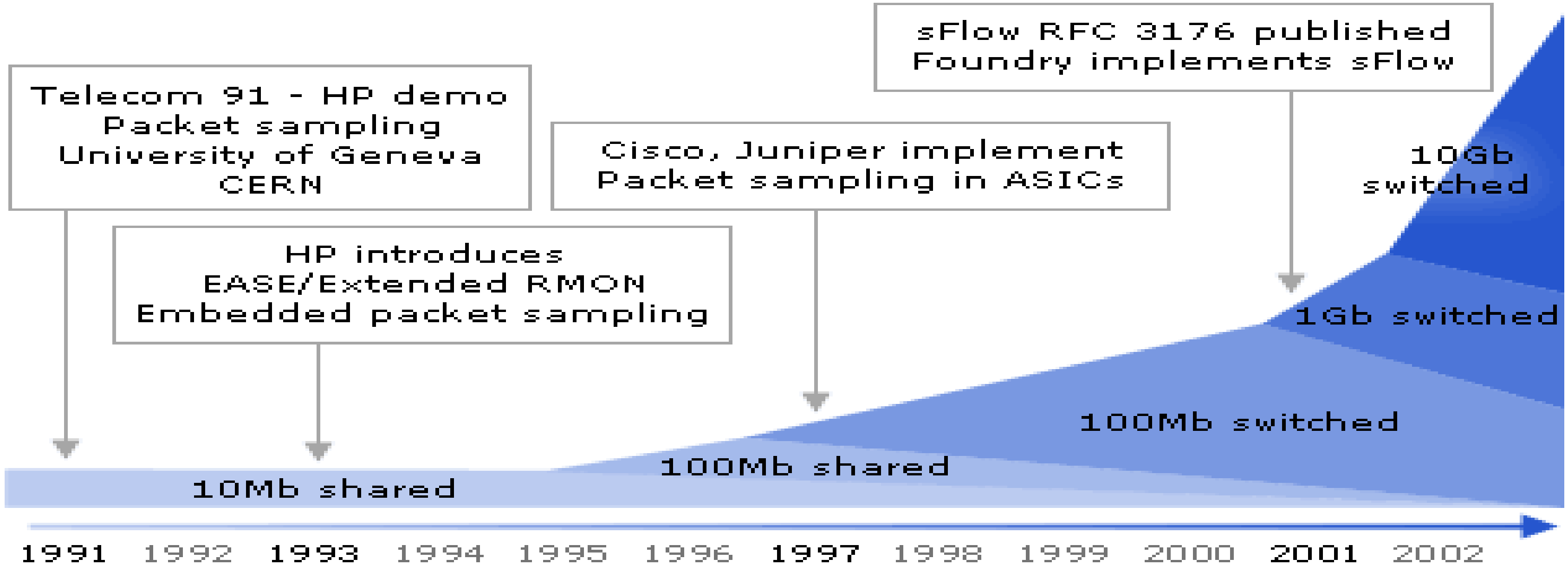
What is sFlow?

- sFlow is a standards based protocol for exporting information about packets traveling through a router or switch, for the purpose of external data analysis.
- Designed by InMon Corporation in 2001.
- Defined (mostly) by RFC 3176.
  - Recent versions have not been published as RFCs.
- Designed to be extremely flexible and extensible.
- Intended to replace/enhance older similar technologies:
  - RMON
  - NetFlow
- Supported by a variety of prominent network gear vendors
  - Foundry, Extreme, Force10, etc.
  - Notably missing: Cisco, Juniper.

A Quick Recap – Why Use Any *Flow?

- External analysis of traffic provides useful info:
  - Protocol, port, and application statistics.
  - DoS tracking and other security monitoring.
  - Traffic analysis for capacity planning.
  - Analysis of traffic over public exchange points.
  - Implementing alternative billing methods.
- Layer 2 analysis can also be used by public exchange point operators to provide peer to peer traffic analysis.
  - Several major exchange points use sFlow for this today
    - Equinix, LINX, AMS-IX, probably others.

The History of Flow Export



- Most popular flow export protocol is NetFlow
  - Developed by Cisco in 1996.
  - Extended several times with new versions/features.
  - Latest version is v9, released in 2004.
- sFlow released in 2001 as an alternative to v5/v8.

So Many Packets, So Little Time

- With the growth of IP traffic, it is simply not practical to look at every packet for analysis.
  - Scaling flow export protocols to work with modern high-speed networks can be done with data consolidation.
- Sampling
  - Only looking at every Nth packet, ex: 1 per 1000 pkts
    - Original data rates can be recreated by multiplication.
    - Reduces amount of data exported and work collecting data.
- Aggregation
  - Combining multiple flow records into a single flow record with less specific information, based on a mask.
    - Reduces data export, but increases work to collect data.
- For efficiency, sampling is usually the winner.

The Problems With NetFlow

- Functions as a true flow-based system
  - Two packets belonging to the same flow must be counted in the same record, which means state must be maintained on the router before data is exported.
  - Flow state memory is frequently subject to resource exhaustion in core networks with many flows.
- Sampling is an afterthought, suffers accordingly
  - Sample rate is not communicated via the protocol.
  - Only one sample rate can be applied for the entire device, and this rate must be configured out of band.
- Support for all versions/features is often sporadic.
- Until v9, unable to export Layer 2 information.

The Features of sFlow

- Designed for high-speed sampling support
  - Not actually flow-based at all, which means no flow state is maintained on the router/switch.
  - Sample rate is communicated in-band with the packet.
  - Different sample rates can be configured on the same host, ex: a lower rate for a 10GE card vs a 100M card.
- Support for flexible message formats
  - Currently 23 types of data-bearing message formats
- Support for Layer 2 information export
- Support for MPLS and BGP route info export
- Support for sampling the complete first 128 bytes
- Support for push-based counter export

More on sFlow Counter Export

- In addition to ordinary “sample” data, sFlow also defines support for push-based “counter” export.
  - This includes data normally polled via SNMP, such as:
    - IfIndex, ifType, ifSpeed, ifStatus, ifInOctets, ifInUCastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifInErrors, etc
  - It turns out to be a very efficient SNMP alternative
    - Eliminates the query portion of the process for data which you know you will always need at set intervals anyways.
    - Eliminates the overhead of SNMP and ASN.1 encoding.
  - This allows interfaces to be easily graphed with very high resolutions, 10-30 seconds instead of 5 minutes.
    - Reveals very interesting traffic patterns.
    - Can be very helpful in quickly reacting to DoS or other events.
    - Doesn't require complex poller code or hurt your router CPU.

The Problems With sFlow

- Mostly protocol-based issues
  - Absolutely mind-boggling header format!
    - Extremely wasteful encoding one minute
      - Using a 32 bit integer for values which will always be “0 – 4”.
    - Unnecessary complexity trying to save space the next minute
      - Separate “compact” message formats which attempt to encode an unrelated 8 bit value onto the end of a field which legitimately uses a 32 bit value, if the value only uses the first 24 bits of space.
  - Complete lack of proper TLV encoding
    - A parser must understand *every* part of every message or it will become desynchronized within the packet.
    - A parser can not skip over sections it does not care about, potentially leading to reduced efficiency.
- Fortunately these issues mostly annoy the developers, not the end users.

NetFlow Fights Back – V9 and IPFIX

- Recognizing the advantages of new features introduced by sFlow, NetFlow v9 was created.
  - Adds Flexible Fields (comparable to sFlow formats)
  - Adds Layer 2 information export capabilities
  - Adds MPLS and BGP information export capabilities
  - Many other major enhancements over old versions
- NetFlow v9 also serves as the basis for “IPFIX”, an IETF standardized flow export protocol.
- Roughly matches most sFlow features.
- The problems:
  - Limited commercial support, currently just Cisco.
  - Doesn't replicate sFlow push-based counters.

Using sFlow

- Many resources at http://www.sflow.org/
- Free Reference Collector by InMon:
  - http://www.inmon.com/technology/sflowTools.php
- sFlow Protocol Specifications
  - http://www.sflow.org/developers/specifications.php
- A tool developed by AMS-IX
  - http://inserturlhere
- A few more items here

Some even cooler tools, honest!

- Coinciding with this presentation, the public release of my high-speed C library, libsflow:
  - http://libsflow.sourceforge.net/
- Advantages:
  - Extremely efficient, handles millions of samples/sec.
  - Portable C library released under BSD license.
- Also comes with reference implementations:
  - A Layer 2 MAC-to-MAC traffic analysis tool.
  - A SNMP-alternative sFlow Counter analysis tool.

Thank You

Name	E-mail Addresses
Richard Steenbergen	ras@nlayer.net