Lossless Compression for IP Flow Information Export (IPFIX)

draft-muenz-ipfix-compression-00

Gerhard Münz, Lothar Braun

72nd IETF Meeting, Dublin, 2008
Starting Point


- **Obvious benefits**
  - less bandwidth required between Exporters and Collectors
  - less network I/O performance and smaller buffer sizes at Exporters and Collectors
  - reduced risk of network congestion ➔ reduced losses

- **What about IPFIX?**
  - bandwidth efficiency not an explicit design goal (not mentioned in RFC3917)
  - yet, we do have mechanisms which help reducing the data volume, e.g.
    - reduced size encoding
    - reducing-redundancy with options
    - biflow export
  - and much more, if information loss is acceptable
    - sampling and filtering
    - aggregation

Experiments with Lossless Compression

► DEFLATE (RFC1951)
  ● LZ77 and Huffman coding
  ● used in various protocols and applications, e.g. IMAP, SIP, gzip, PNG

► Applied to Data Sets of different size
  ● 10, 20, 40, 60, and 100 Data Records
  ● ZLIB compressed data format (RFC1950)
    ➤ header (2 bytes) and CRC trailer (4 bytes)

► Measurement data
  ● Flow Records from two networks
    ➤ university network
    ➤ ISP backbone
  ● Packet Reports
    ➤ network of a research institute


**Lossless Compression for IP Flow Information Export (IPFIX)**

**Compression of Flow Records**

Mean compression ratio from 6.8 (10 records) to 46 (100 records)

**Template:**

<table>
<thead>
<tr>
<th>Field No</th>
<th>Information Element</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sourceIPv4Address</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>destinationIPv4Address</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>sourceTransportPort</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>destinationTransportPort</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>protocolIdentifier</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>octetDeltaCount (*)</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>packetDeltaCount (*)</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>flowStartSeconds</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>flowEndSeconds</td>
<td>4</td>
</tr>
</tbody>
</table>

(*) reduced size encoding

Each additional Record costs 0.22 Bytes only!
Compression of Packet Reports containing Packet Sections

Template:

<table>
<thead>
<tr>
<th>Field No</th>
<th>Information Element</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>observationTimeSeconds</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>ipHeaderPacketSection (*)</td>
<td>variable</td>
</tr>
</tbody>
</table>

(*) IP and transport layer

Mean compression ratio from 1.27 (10 records) to 1.76 (100 records)
Realization of IPFIX Compression

- Compressed IPSec tunnel using IPComp (RFC3173)
  - compression at IP datagram level

- TLS/DTLS compression (RFC3749)
  - compression at level of SCTP messages, UDP datagrams, TCP stream

- IPFIX extension I: Compressed IPFIX Message
  - IPFIX Header (version=11)
  - compressed message payload (Sets)

- IPFIX extension II: Compressed Data Sets
  - Deflate Template Set (same structure as normal Template Set)
  - corresponding Data Set containing compressed Data Records
Conclusion

➤ **Significant compressibility of Data Records**
  - very high for typical Flow Records
  - much lower for Packet Reports containing packet sections

➤ **Possible realizations based on network and transport layer**
  - IPSec tunnel
  - TLS/DTLS (mandatory according to RFC5101!)

➤ **We can think of more flexible IPFIX-specific solutions**

➤ **Not sure, if this is an IPFIX standardization issue**

Thank you.