Enhanced Netflow Version 9 (e-Netflow v9) For Network Mediation: Structure, Experiment and Analysis

Mohd Saufy Rohmad 1,2, Farok Azmat 1, Mazani Manaf 1

Network Flow Measurement Test bed,
1Computer Technology and Networking Department,
Faculty of Information Technology and Quantitative Science, UiTM, Malaysia
mazani@tmsk.uitm.edu.my, farok@tmsk.uitm.edu.my
2Researcher, SmartwareLab, Shah Alam, Malaysia
smartwarelab@yahoo.co.uk

Abstract

The Netflow version 9 is the latest Netflow version introduced by CISCO for the network monitoring and analysis purposes. Then come IPFIX that defined by IETF to overcome problem in Netflow version 9 and to enhance a few aspect of network flow capturing and filtering. It co-operated with PSAMP (packet sampling) standard for intelligent packet sampling for flow monitoring. This research paper will introduce enhanced Netflow version 9 (e-Netflow v9) that use nProbe GPL tool. The enhanced Netflow version 9 uses existing Netflow v9 parameters with RTP and SIP plug-ins. We will introduce the overall structure of this Netflow, the experiment to compare the results and performance and the finally analysis parts to study the reliability of this e-Netflow v9.

Key Words: Netflow, Collector, Mediation, nProbe

1. Introduction

Netflow is widely used by researchers and network administrators to monitor the network usage [6]. The most widely use is Netflow version 5 that use static Netflow format.[7] There are many available GPL tools that process, analyzed and present Netflow version 5 format. However, with Netflow v9 format, there are still less GPL tools available on the web. Besides that, the dynamic usage of Netflow v9 is less explore by the researchers to produce advanced monitoring system for all range of network services.

With the GPL tool nProbe[1], we enhanced the existing Netflow v9 available parameters with a few additional attributes with SIP and RTP plug-in. We define this e-Netflow v9 for the general purpose mediation system. General purpose mediation system means that we are collecting as much as information from the network and store every single flow information in e-Netflow v9 UDP packet.

This paper is organized as follows:
Part 1 : Introduction
Part 2 : e-Netflow v9 Format and Structure
Part 3 : Experiment Setup
Part 4 : Experiment Results
Part 5 : Result comparison with Netflow v5
Part 6 : e-Netflow v9 Usage and Applications
Part 7 : Finding, Analysis and Discussion
Part 8 : Conclusion and Future Enhancement
Part 9 : Acknowledgement
Part 10 : Reference

2. e-Netflow v9 Format and Structure

e-Netflow v9 used completely same attributes and structure of cisco Netflow version 9 except little enhancement of RTP and SIP plug-in. This enhancement is based on the contribution by developer in nProbe [2] GPL software.

Below we list all the capture attributes that we define in e-Netflow v9 format[3]. We divide the information to three categories. The first one is general monitoring parameters that include Bytes number in and out, number of flows, protocol number, Type of Service, TCP Flags, Port number, Input and Output index and so on. We can se directly from the attributes name what its definitions. This attributes is the core attributes that can be used for network mediation system. We define maximum number of attributes for the ease of higher level applications.

General Attributes
%IN_BYTES %IN_PKTS %FLOWS %PROTOCOL
%SRC_TOS %TCP_FLAGS %L4_SRC_PORT
%IPV4_SRC_ADDR %SRC_MASK %INPUT_SNMP
%L4_DST_PORT %IPV4_DST_ADDR %DST_MASK
%OUTPUT_SNMP %IPV4_NEXT_HOP %SRC_AS %DST_AS
%LAST_SWITCHED %FIRST_SWITCHED
%IN_DST_MAC %OUT_SRC_MAC %FRAGMENTED
%ICMP_TYPE

Flow Capture parameters holds information regarding the capturing process. The sampling interval shows the
In this paper, we focus on the monitoring of RTP and SIP protocols. RTP is Real Time Protocol, used for multimedia streaming. SIP is Session Initiation Protocol, used for creating, managing, and terminating multimedia sessions. The monitoring of these protocols is crucial for network management and troubleshooting.

**Flow Capture Parameters**

- **SAMPLING_INTERVAL**
- **SAMPLING_ALGORITHM**
- **FLOW_ACTIVE_TIMEOUT**
- **FLOW_INACTIVE_TIMEOUT**
- **ENGINE_TYPE**
- **ENGINE_ID**

**MPLS Label**

MPLS (Multiprotocol Label Switching) is a protocol that allows network operators to provide multiple services over a single backbone infrastructure. MPLS Label (10 attributes) is special parameters under Netflow v9 for the monitoring of MPLS label.[4] This is very useful for MPLS enable networks and for Internet Service Provider.

**SIP Attributes**

- **SIP_CALL_ID**
- **SIP_CALLING_PARTY**
- **SIP_RTP_SRC_PORT**
- **SIP_RTP_DST_PORT**

**RTP Attributes**

- **RTP_FIRST_SSRC**
- **RTP_LAST_SSRC**
- **RTP_OUT_PAYLOAD_TYPE**

Below is a sample of XML files that describe the netflow parameters.

```xml
<?xml version="1.0" standalone="yes" encoding="UTF-8"?>
<NetFlow_Packet from="a0a0a01">
  <NetFlow_Header>
    <Netflow Version Number = 9>
      <Version length="2">0009</Version>
      <H>Netflow header number</H>
      <Count length="2">0001</Count>
      <System_Uptime length="4">0001246</System_Uptime>
      <Unix Seconds length="4">1169976805</Unix Seconds>
      <Packet Sequence Number length="4">00000018</Packet Sequence Number>
      <Source ID length="4">00000000</Source ID>
      <Closing Tags of Netflow Header>
        <NetFlow_Header/>
        <Opening Tags of Netflow Records of Information>
          <NetFlow_Record>
            <Netflow UDP Packet Header>
              <Source Network Mask length="4">0a8647f</Source Network Mask>
              <Destination Network Mask length="4">5ae0</Destination Network Mask>
              <Input Interface Index length="2">5ae0</Input_Interface_Index>
              <Destination Port Number length="2">9595</Destination Port Number>
              <IPv4 Source Address length="4">0436</IPv4 Source Address>
              <IPv4_DST_ADDR length="4">00000000</IPv4_DST_ADDR>
              <Output Index length="2">00</Output Index>
            </Netflow_UDP_Packet_Header>
            <Flow Active Time to be Captured by nProbe>
              <Flow Active Timeout specifications>
                <Flow Active Timeout length="2">0436</Flow Active Timeout>
              </Flow Active Timeout specifications>
              <Flow Inactive Timeout specifications>
                <Flow Inactive Timeout length="2">0436</Flow Inactive Timeout>
              </Flow Inactive Timeout specifications>
              <Output Index specifications>
                <Output Index length="2">00</Output Index>
              </Output Index specifications>
              <Flow Source Engine ID>
                <Engine ID length="4">0436</Engine ID>
              </Flow Source Engine ID>
              <Flow Source Engine Type specification>
                <Engine Type length="4">0436</Engine Type>
              </Flow Source Engine Type specification>
              <Flow Source Engine Information>
                <Engine Information length="4">0436</Engine Information>
              </Flow Source Engine Information>
            </Flow Captured by nProbe>
          </NetFlow_Record>
        </Opening Tags of Netflow Records of Information>
      </Closing Tags of Netflow Header>
    </Netflow Version Number = 9>
  </NetFlow_Header>
</NetFlow_Packet from="a0a0a01">
```

After the experiment conducted using pre-capture traffic using tcpdump, we analyzed the traffic using nProbe to generate netflow UDP packet. Then the XML files are generated from the conversion. Below is the sample of XML files of e-Netflow v9 for network mediation purposes. The details of each attributes and parameters will explain with the XML file. The # sign is point to comment that describes the netflow parameters.

# This is the netflow version that defined in each xml file

```xml
<?xml version="1.0" standalone="yes" encoding="UTF-8"?>
<NetFlow_Packet from="a0a0a01">
  <NetFlow_Header>
    <Netflow Version Number = 9>
      <Version length="2">0009</Version>
      <H>Netflow header number</H>
      <Count length="2">0001</Count>
      <System_Uptime length="4">0001246</System_Uptime>
      <Unix Seconds length="4">1169976805</Unix Seconds>
      <Packet Sequence Number length="4">00000018</Packet Sequence Number>
      <Source ID length="4">00000000</Source ID>
      <Closing Tags of Netflow Header>
        <NetFlow_Header/>
        <Opening Tags of Netflow Records of Information>
          <NetFlow_Record>
            <Netflow UDP Packet Header>
              <Source Network Mask length="4">0a8647f</Source Network Mask>
              <Destination Network Mask length="4">5ae0</Destination Network Mask>
              <Input Interface Index length="2">5ae0</Input_Interface_Index>
              <Destination Port Number length="2">9595</Destination Port Number>
              <IPv4 Source Address length="4">0436</IPv4 Source Address>
              <IPv4_DST_ADDR length="4">00000000</IPv4_DST_ADDR>
              <Output Index length="2">00</Output Index>
            </Netflow_UDP_Packet_Header>
            <Flow Active Time to be Captured by nProbe>
              <Flow Active Timeout specifications>
                <Flow Active Timeout length="2">0436</Flow Active Timeout>
              </Flow Active Timeout specifications>
              <Flow Inactive Timeout specifications>
                <Flow Inactive Timeout length="2">0436</Flow Inactive Timeout>
              </Flow Inactive Timeout specifications>
              <Output Index specifications>
                <Output Index length="2">00</Output Index>
              </Output Index specifications>
              <Flow Source Engine ID>
                <Engine ID length="4">0436</Engine ID>
              </Flow Source Engine ID>
              <Flow Source Engine Type specification>
                <Engine Type length="4">0436</Engine Type>
              </Flow Source Engine Type specification>
              <Flow Source Engine Information>
                <Engine Information length="4">0436</Engine Information>
              </Flow Source Engine Information>
            </Flow Captured by nProbe>
          </NetFlow_Record>
        </Opening Tags of Netflow Records of Information>
      </Closing Tags of Netflow Header>
    </Netflow Version Number = 9>
  </NetFlow_Header>
</NetFlow_Packet from="a0a0a01">
```
We conduct this experiment in our Network Flow Measurement Test bed using 2 Linux PC that used Fedora Core 4 with Pentium 4 2.8 GHz and 512DDR4 RAM.

We captured 4.3Gb traffic using Omnipack with the helps of NetOptics zero delay network tap for capturing accuracy. This traffic is used as our basis and comprison benchmark with other Netflow format. Then the omnipeak format is being converted to tcpdump format using wireshark tool.

The captured traffic that consists of 18 files that size 200Mb each is stored to our first machine. Then nProbe is run on this machine with the direction to calculate e-Netflow v9 with pre-captured traffic.

Then the nProbe is instructed to send the filtered e-Netflow v9 packet to second machine that run netflow2XML applications. Netflow2XML is run to capture binary format netflow on specific port, convert to XML format and store it to directory. Then the XML files are analyzed for result presentation and analysis. Figure below show our setup in diagram.

![Experiment Setup for e-Netflow v9 Testing](image)

**Figure 2: Experiment Setup for e-Netflow v9 Testing**

The same experiment setup is used for other netflow experiment including well known netflow v5 format In further chapter we will compare the results from e-Netflow v9 with Netflow v5 results.

In this experiment test the individual files e-Netflow v9 generations 10 times for result validity. The simple statistical analysis is done to the result to calculate mean, standard deviation and average mean of the results. Highest and lowest percentage also calculated to compare the highest and lowest peak gap.

### 4. Experiment Results

10 times experiments for the same files produce strong results. Due to the structure of nProbe and netflow2xml that use a lot of buffer, the result stability is reduce. Moreover the e-Netflow v9 format use template based communications with collector make the collector need to
perform extra work on flow receiving and identifications. This is prove in following part when we compare the output and result between e-Netflow v9 with Netflow v5. Netflow v5 will show higher and stable output compare to e-Netflow v9.

Above simple calculations show how the percentage of record is calculated. This will show the result stability of e-Netflow v9 statistics.

\[
\%\text{diff ratio} = \frac{r_{\text{highest}} - r_{\text{lowest}}}{r_{\text{highest}}} \times 100
\]

Above figure show how the result of e-Netflow v9 varied in term of the highest-lowest stability. The most stable result occurs in capture 15 and 12 when the different percentage only 19-20% and the standard deviation is 46.42 and 51.04 respectively. We can see the results are mostly unstable and unpredictable. This show for the high speed and unstable traffic, e-Netflow v9 usage is not suitable and the information loss may occur.

![Figure 3: Output Stability of e-Netflow v9](image)

Figure above show the total results of this experiment. We can see the high rate of fluctuation of highest peak of each session. The highest peak from all session recorded in session 18 where the xml files produce more than 1800 files. However the lowest number of files generated is stable around 600-700 files from session 4 and above. This show the minimum capability of nProbe and netflow2XML process e-Netflow v9 format. This also show that is the minimum files nProbe can produce. It means that the nProbe maximum stress pushing can go only up to 500 flows in 200Mb tcpdump file.

Next, we will compare the result of e-Netflow v9 with Netflow v5 format. This comparison is very crucial for performance comparison and benchmarking. This also can be used for monitoring prediction for financial oriented applications like usage based billing.

5. Result comparison with Netflow v5

As stated before, this comparison is very essential to benchmark the performance of e-Netflow v9. From all our experiment, Netflow v5 produce very stable and high throughput of xml files. Compare to netflow v9, Netflow v5 used template for probe and collector process. This will introduce extra overhead in memory buffer and bytes used.
Above figure show the overall graph for Netflow v5 experiment. We can see the results are much more stable. The standard deviations lines show the stable line compare to previous e-Netflow v9. The highest xml recorded also high, near 6000 files generated on session number 13. This shows the Netflow v5 produce 4 times more than e-Netflow v9. This is rough statement that needs to be further research.

![Figure 5 : Result Summary Graph for Netflow v5](image)

**Figure 5 : Result Summary Graph for Netflow v5**

Above figure show the difference ratio of highest and lowest records of Netflow v5. It show the stable and high throughput difference to e-Netflow v9 statistic shown before. It tell us that although Netflow v5 produce good results on highest records, the difference between high and low records is very high. And the high ratio is stable for most of the session. It show that the Netflow v5 produce stable but very bad results in term of traffic gap compare to e-Netflow v9 that produce more low standard deviations.

![Figure 6 : Difference Ratio for Netflow v5](image)

**Figure 6 : Difference Ratio for Netflow v5**

6. **e-Netflow v9 Usage and Applications**

From the starting of this paper maybe the reader is thinking about the applications that can be use on the top if this network mediation idea.

1. **Usage Based Billing**

From the general attributes itself, the usage based billing applications can be develop on top of this mediation architecture. This application only uses a few of the main attributes including source and destination port and address, protocol and bytes transferred.

2. **Security Monitoring**

From the source-destination port and address, protocol, bytes, packet and TCP Flags, security monitoring can be performed. This includes the SYN flags storm monitoring and other DoS attack. Other attack such as ICMP and SMTP relay also can be detected.

3. **Link Utilization Calculations**

From the e-Netflow v9, we simply can calculate the current link utilization compare to total available bandwidth. This calculation involved time based flow generation and termination with the number of bytes it transferred.

4. **Traffic Forecast and Engineering**

Traffic forecast involved the intelligent analysis of Netflow with the future prediction of traffic trends. It involve mathematical calculations and detail analysis of Netflow attributes.

5. **Host identifications**

With e-NEtflow v9 that capable for detecting host fingerprint, we can detect the fingerprint for each host that transmits the packets.

6. **Sampling Information Analysis**

Sampling information can be used for information and consideration from the network administrator. This is very crucial for mission critical monitoring that need extra accurate per packet and per flow information. This also can be used for measurement of original real traffic from the captured traffic.

7. **Flow Authentications**

Flow Authentications mean that this e-Netflow v9 can be used to authenticate the original flow between the collector and source. It’s due to the ability of nProbe to set the Engine ID for the flow source. It prevents the fraud in duplicating flow source.

Above are applications that can be develops on top of this e-Netflow v9 mediations architecture.

7. **Finding, Analysis and Discussion**

This research and experiment tell us that the usage of Netflow v9 or template based network flow generation will introduce extra processing overhead to the CPU clock and memory.

This experiment also shows the process of converting binary data to XML will use extra resource to the network and computer and will cause extra delay.
Netflow monitoring in high speed network need high speed packet capture and high speed packet filter for reliable Netflow processing capabilities.

e-Netflow v9 throughput is less 4 times than ordinary Netflow v5 results. This rough formula can be the basis for further e-Netflow deployment.

Each attributes in the netflow play role in adding buffer load for traffic filtering. This shows that the less and accurate e-Netflow v9 configurations will prevent the nProbe to perform slowly.

8. Conclusion and Future Enhancement

This research shows the preliminary result of the testing and analysis of what we define as e-Netflow v9. Although this research show the basic usage and experiment of e-Netflow v9, stronger and depth experiment is needed.

Currently we are focusing more on traffic engineering analysis for detail netflow analysis. For current and future enhancement we will focus on below area:

1. Traffic Generator is used to generate traffic instead of Pre-capture traffic is used for accurate traffic measurement and Netflow data comparison.


3. nProbe modifications is needed to handle high speed data with exact flow to collector synchronization. This synchronization means that the template based communications is eliminated to direct parser to the collector.

4. Configurable interface for nProbe and nProbe collector also can be develops for flexible usage of portable Netflow. The GUI interface using gtk+ or PHP is very useful to reduce difficulty in handling nProbe from shell interface.

Finally, we are very welcome and glad if any industrial based company, research institutions or government research body interested to fund the enhancement of this research.

We also ready for further discussions and research opportunity related to Network flow measurement and monitoring.

9. Acknowledgments

We want to acknowledge our colleagues from Network Protocol Unit, Telekom Malaysia R&D for the opportunity to pursue research with them. We also thank Department of CTN for PC and server for this project test-bed. And congratulations to all the team at TM R&D for Usage based billing system implementation at Multi-net, Pakistan that use Netflow v9 for mediation system.

10. References


