Software Tool for Passive Real-Time Measurement of QoS Parameters

Mihai Vlad, Ionut Sandu, Alcatel Timisoara, Romania

Virgil Dobrota, Ionut Trestian, T.U. of Cluj-Napoca, Romania

Jordi Domingo-Pascual, U.P.C. Barcelona, Spain
1. Introduction
2. QoS Parameters
   OWD, AOWD, IPDV, AIPDV
3. Designing of the Measurement Tool
   Packet capturing
   Time-stamping
   Flow ID generation
   Classification
   Packet ID generation
   Measurement result transfer
4. Proposed Architecture
5. Experimental Results
6. Conclusions
1. Introduction (I)

- New QoS standards for Heterogeneous Networks: EuQoS – FP6 European Project

- Developed at Universitat Politècnica de Catalunya, Barcelona (Socrates-Erasmus grant in 2005)

- Based on OreNETa (One-way delay REaltime NETwork Analyzer) – Abel Navaro

- Real-time network analysis
  - QoS parameters computed on-the-fly as average values over a period of time (e.g. one second)
Passive capturing advantages
- No additional traffic on the network
- Packets are not modified

Heterogeneous Networks: different transmission technologies between two flow endpoints
- Ethernet/IEEE 802.3
- Fast Ethernet/IEEE 802.3u
- Wireless LAN (802.11b/g)
- ATM
- Gigabit Ethernet
2. QoS Parameters: (Average) One Way Delay

\[ OWD_i = t_{1i} - t_{0i} \quad 1 \leq i \leq N \]

\[ AOWD = \frac{\sum_{i=1}^{N} OWD_i}{N} \]
2. QoS Parameters: IP Packet Delay Variation (I)

\[ IPDV_i = OWD(i-1) - OWD_i \]
2. QoS Parameters: IP Packet Delay Variation (II)
3. Designing of the Measurement Tool (I)

**Flows:** defined by the *five-tuple*
- IPv4/IPv6 Source Address
- IPv4/IPv6 Destination Address
- Source Port
- Destination Port
- Transport Layer Protocol

<table>
<thead>
<tr>
<th>IPv4 Source Address</th>
<th>IPv4 Destination Address</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Protocol Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>63</td>
<td>79</td>
<td>95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IPv6 Source Address</th>
<th>IPv6 Destination Address</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Protocol Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>127</td>
<td>255</td>
<td>271</td>
<td>287</td>
</tr>
</tbody>
</table>
3. Designing of the Measurement Tool (II)

Diagram:

- **Packet Capturing** → **Timestamping** → **Flow ID Generation** → **Classification** → **Packet ID Generation** → **Measurement Result Transfer** → **Analyzer** → **QoS Parameters Processing** → **Measurement Result Transfer**
3. Designing of the Measurement Tool (III)

- **Packet capturing**
  - `pcap` library (libpcap 0.8.3 or later)

- **Time-stamping**
  - Capturing points must be synchronized
  - *GPS (Global Positioning System)*
  - *NTP (Network Time Protocol)*

- **Flow ID**
  - CRC-32 on the *five-tuple*

- **Classification**
  - Each packet is assigned a sequence number depending on the flow it belongs to

- **Packet header** fields
  - Must not vary during the transport through the network
  - Should be highly variable between different packets

- **Data** fields
  - first 27 bytes of payload
3. Designing of the Measurement Tool (IV)

**Binary File**

<table>
<thead>
<tr>
<th>crc_packet</th>
<th>seq nr</th>
<th>sec</th>
<th>usec</th>
</tr>
</thead>
<tbody>
<tr>
<td>03 00 47 D7 9B D6 02 00 00 00 42 93 67 7A 00 0E 20 ED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6F 0B 03 00 F6 BE 03 91 03 00 00 00 42 93 67 7A 00 0E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 0E BD 2C 03 00 46 28 9A 92 04 00 00 00 42 93</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67 7A 00 0F 0B 4B 03 00 6B A0 28 E8 05 00 00 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 00 68 99 5C 24 06 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 00 42 93 67 7B 00 00 65 53 03 00 B8 83 81 F9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 B3 62 03 00 D1 EF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 00 68 99 5C 24 06 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 00 42 93 67 7B 00 00 65 53 03 00 B8 83 81 F9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 B3 62 03 00 D1 EF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 00 68 99 5C 24 06 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 00 42 93 67 7B 00 00 65 53 03 00 B8 83 81 F9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 B3 62 03 00 D1 EF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 00 68 99 5C 24 06 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 00 42 93 67 7B 00 00 65 53 03 00 B8 83 81 F9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 B3 62 03 00 D1 EF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 00 00 00 42 93 67 7B 00 00 00 68 99 5C 24 06 00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3. Designing of the Measurement Tool (V)

#### Aggregate File

<table>
<thead>
<tr>
<th>Flow</th>
<th>Seq</th>
<th>Src</th>
<th>Dest</th>
<th>Tx Time</th>
<th>Rx Time</th>
<th>Size</th>
</tr>
</thead>
</table>
4. Proposed Architecture (1)

Traffic capture and classification

Processing of QoS parameters perflow

Real-time presentation of QoS parameters

Data Store

Legend:
- Control
- Data

Passive measurement
4. Proposed Architecture (II)

**OReNETa’s Packet Reports**

- **IPv4:** 14-byte redundancy (14+14=28 bytes)

<table>
<thead>
<tr>
<th>L3 Protocol</th>
<th>sec</th>
<th>usec</th>
<th>crc</th>
<th>IPv4 Length</th>
<th>IPv4 Src Address</th>
<th>IPv4 Dst Address</th>
<th>L4 Protocol</th>
<th>L4 Src Port</th>
<th>L4 Dst Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>15</td>
<td>19</td>
<td>23</td>
<td>24</td>
</tr>
</tbody>
</table>

- **IPv6:** 38-byte redundancy (14+38=52 bytes)

<table>
<thead>
<tr>
<th>L3 Protocol</th>
<th>sec</th>
<th>usec</th>
<th>crc</th>
<th>IPv6 Length</th>
<th>IPv6 Src Address</th>
<th>IPv6 Dst Address</th>
<th>L4 Protocol</th>
<th>L4 Src Port</th>
<th>L4 Dst Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>31</td>
<td>47</td>
<td>48</td>
<td>50</td>
</tr>
</tbody>
</table>

4. Proposed Architecture (III)

**New Approach**

- Flow Descriptors (sent at the beginning of capture)
- Headers (23 bytes instead of 28/52 bytes)
4. Proposed Architecture (IV)
4. Proposed Architecture (V)

**Meter:**
- Packet Capturing (use *libpcap*)
- Timestamping (use *libpcap*)
- Flow ID Generation (CRC-32 of the five-tuple)
- Classification (generate the binary tree of flows – and send *flow descriptor* or *header*)
- Packet ID Generation (compute CRC-32 for highly variable data in the packet)
- Send the measurement results to the Analyzer
4. Proposed Architecture (VI)
4. Proposed Architecture (VII)

Analyzer:
- Connect to the *meters* and command them through the communication protocol
- Fetch the messages from the *meters* (*flow descriptors or headers*) and handle them
- Route the *headers* to the appropriate flow structures (binary tree)
- Synchronize the received *headers* to compute the QoS parameters
- Present the data to Graphical clients
- Handle interactive commands from the command-line
4. Proposed Architecture (VIII)

Analyzer
Basic Architecture


Legend

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>1</td>
<td>Analyzer</td>
</tr>
<tr>
<td>🔄</td>
<td>2</td>
<td>Meter</td>
</tr>
<tr>
<td>🔄</td>
<td>2</td>
<td>Router</td>
</tr>
<tr>
<td>🔄</td>
<td>3</td>
<td>PC</td>
</tr>
</tbody>
</table>
Testbed - 2 Capture Points

Source

Portatil

192.168.13.2 (eth1) → 192.168.13.1 (eth1) → 192.168.11.3 (eth0) → 192.168.11.4 (eth1) → Verdi

802.11b

Analyzer

147.83.130.176 (eth0)

Fast Ethernet

Pampol

192.168.11.1 (FastEthernet0/1)

Fast Ethernet

Vinyater

130.206.212.50 (ATM2/0.6)

ATM

Malvasia

192.168.31.1 (eth0) → 192.168.31.2 (eth0) → 192.168.30.1 (eth0) → 192.168.30.2 (eth0) → Sunnoli

Destination

Presario

Offline Results from 2 *meters*

- 30 seconds MGEN traffic from *Portatif* to *Presario*, with *meters* running on *Verdi* and *Malvasia*:

  mgen -i eth0 -b 192.168.31.2 -s 370 2000000 -r 100

- **Analyzer** command-line:

  ./analyzer -l
  -i 147.83.130.164 1 -d eth0 1 -w out 1
  -i 147.83.130.171 2 -d eth1 2 -w out 2
  -f "udp"
OWD for 3000 packets

One-Way delay

NETMETER

packet sequence number
OWD Distribution for 3000 packets

One-Way delay distribution

% of packets

NETMETER

IPDV Distribution for 3000 packets
Testbed - 4 Capture Points

192.168.13.2 (eth1) -> 192.168.13.1 (eth1) -> 192.168.11.3 (eth0) -> 192.168.11.4 (eth1)

802.11 b

Fast Ethernet

192.168.11.1 (FastEthernet0/1)

130.206.212.50 (ATM2/0.6)

130.206.212.49 (ATM2/0.6)

Real-time Testing with 4 meters

MGEN traffic from *Portatil* to *Presario*, with *meters* running on *Pampol, Verdi* (in and out) and *Malvasia*:

```
mgen -i eth0 -b 192.168.31.2 -s 512 2000000 -r 200
```

*Analyzer* command-line:

```
./analyzer
   -i 147.83.130.163 1  -d eth0 1  -w out 1
   -i 147.83.130.171 2  -d eth1 2  -w out 2
   -i 147.83.130.164 3  -d eth0 3  -w out 3
   -i 147.83.130.171 4  -d eth1 4  -w in  4
   -f "udp and host 192.168.31.2"
```
### Results - Real-time Screen Capture

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MTR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPS</td>
<td>200 packets/s</td>
<td>200 packets/s</td>
<td>200 packets/s</td>
</tr>
<tr>
<td>THROUGHPUT</td>
<td>108000 bytes/s</td>
<td>108000 bytes/s</td>
<td>108000 bytes/s</td>
</tr>
<tr>
<td>OWD</td>
<td>0.001656 seconds</td>
<td>0.000210 seconds</td>
<td>0.007587 seconds</td>
</tr>
<tr>
<td>IPDV</td>
<td>0.000000 seconds</td>
<td>-0.000000 seconds</td>
<td>0.000000 seconds</td>
</tr>
<tr>
<td>PKLOSS</td>
<td>0 packets/s</td>
<td>0 packets/s</td>
<td>0 packets/s</td>
</tr>
</tbody>
</table>

Conclusions

- Improvements, optimizations and additional functionalities compared to original OReNETa
- Flow descriptors and headers optimize the traffic between the meter and the analyzer. The values identifying a flow (five-tuple) are only sent with the flow descriptor.
- Size of packet reports: reduced from 28 to 23 bytes (IPv4), and from 52 to 23 bytes (IPv6).
- Irrelevant packets (e.g. ICMP and ARP) were discarded at the meter
- More than one analyzer to a meter and vice-versa
- Pure binary files to store only the needed data, and the processing is performed later, when the capture is finished.