Simulating Network Link Compression in Loss-less Wireless Sensor Networks (WSNs) Environment

Team: Rozita Teymourzadeh, Ramona Carmen Stoic, Robert Chang, Vahab Pournaghshband

University of San Francisco
Motivation

More and more industries adapt to WSNs and generate WSN node in the network. Loss-less data compression is essential solution to handle the high data traffic produced by WSNs. Hence, simulating loss-less compression link in open-source Network Simulator 3 (NS3) and validate the simulated environment motivate us to implement this research project.
Objective

- To simulate network link compression in Loss-less Wireless Sensor Networks (WSNs)
- To compress network link using zLib library
- To validate the simulated environment
Terms:

High Entropy:

High Entropy traffics is a wide verity of values between 0x00 and 0xFF. It may consist of uniform distribution and dump data.

Low Entropy:

Low Entropy traffics is a limited range of values and very skewed distribution of data and limited ASCII characters.
LZS Compression

- Lempel–Ziv–Stac is a lossless data compression algorithm
- Combination of the LZ77 sliding-window compression and fixed Huffman coding.

<table>
<thead>
<tr>
<th>Length</th>
<th>Bit encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td>3</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>1100</td>
</tr>
<tr>
<td>6</td>
<td>1101</td>
</tr>
<tr>
<td>7</td>
<td>1110</td>
</tr>
<tr>
<td>8 to 22</td>
<td>1111 xxxx, where xxxx is length – 8</td>
</tr>
<tr>
<td>23 to 37</td>
<td>1111 1111 xxxx, where xxxx is length – 23</td>
</tr>
<tr>
<td>length &gt; 37</td>
<td>(1111 repeated N times) xxxx, where N is integer result of (length + 7) / 15, and xxxx is length - (N*15 - 7)</td>
</tr>
</tbody>
</table>

Ref. Cisco
System Architecture

PointToPointNetDevice::Send()

High Entropy
S1

Low Entropy
S2

R1

isCompress()
packet->CopyData(&inData[2]),size

Compress

R1

packet->RemoveHeader(header)

PointToPointNetDevice::PppToEther (uint16_t proto)

R1

PointToPointNetDevice::Send()
System Architecture

![Diagram showing system architecture with nodes S1, S2, and R1 connected with arrows labeled High Entropy and Low Entropy.]
System Architecture

S1

High Entropy

PointToPointNetDevice::Send()

S2

R1

Low Entropy

isCompress()? NO

R1

R1
System Architecture

PointToPointNetDevice::Send()

packet-> CopyData(&inData[2]),size)

isCompress()? YES isDecompress()? NO

packet->RemoveHeader(header)
PointToPointNetDevice::PppToEther (uint16_t proto)
PointToPointNetDevice::Send()
System Architecture

PointToPointNetDevice::Send() packet-> CopyData(&(inData[2]),size)
PointToPointNetDevice::Receive()
isCompress()? NO isDecompress()? YES
packet->RemoveHeader(header)
PointToPointNetDevice::PppToEther (uint16_t proto)
System Architecture

PointToNetDevice::Receive()

packet->RemoveHeader(header)
PointToNetDevice::PppToEther (uint16_t proto)
PointToNetDevice::Receive()
System Architecture

Packet arrival

Packet Departure
System Design

NS3:

```
.udp-client.cc
.udp-server.cc
.point-to-point-net-device.cc
.point-to-point.cc (application)

.AddAttribute("Entropy","Boolean Value", BooleanValue(true),
MakeBooleanAccessor(&UdpClient::m_entropy),
MakeBooleanChecker());
```

```cpp
bool isHighEntropy; /* entropy flag */

bool doCompress = false; /* Compress flag */
```
System Design

- Configuration Management System
- Compression System
- Topology Management System
- Synchronization and output recorder
Detection Tool

\[
\Delta t^H_{\text{HighEntropy}} = T_{\text{First Pkt Arrival}} - T_{\text{Last Pkt Arrival}}
\]

\[
\Delta t^L_{\text{LowEntropy}} = T_{\text{First Pkt Arrival}} - T_{\text{Last Pkt Arrival}}
\]

**Detection Factor** = \( \Delta t^H - \Delta t^L \)
/*
  * To compile and build the project
  */
$ cd workspace/Transport-Layer-Security
/ns-3-allinone/ns-3-dev
$ ./waf configure
$ ./waf build
$ ./waf --run "scratch/point2point
--IsHighEntropy=1
--IsCompress=1
--MaxPacketCount=2"
Validation

Wireshark Packet Analyzer
Validation

Compression in capacity link bottle-neck
Validation

High & Low Packets Arrival Time

- High Entropy Packet Train
- Low Entropy Packet Train
Conclusion

- Enriched open-source NS-3 simulation environment with compression feature
- Simulated packet compression algorithm when the network faced with high traffic.
- Testing and validation of compression and decompression embedded model.
- Result shows Compression has no effect on the data transmission speed if there is no bottle-neck exists in the capacity compression link.
Reference

