



UNIVERSITY OF
SAN FRANCISCO

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

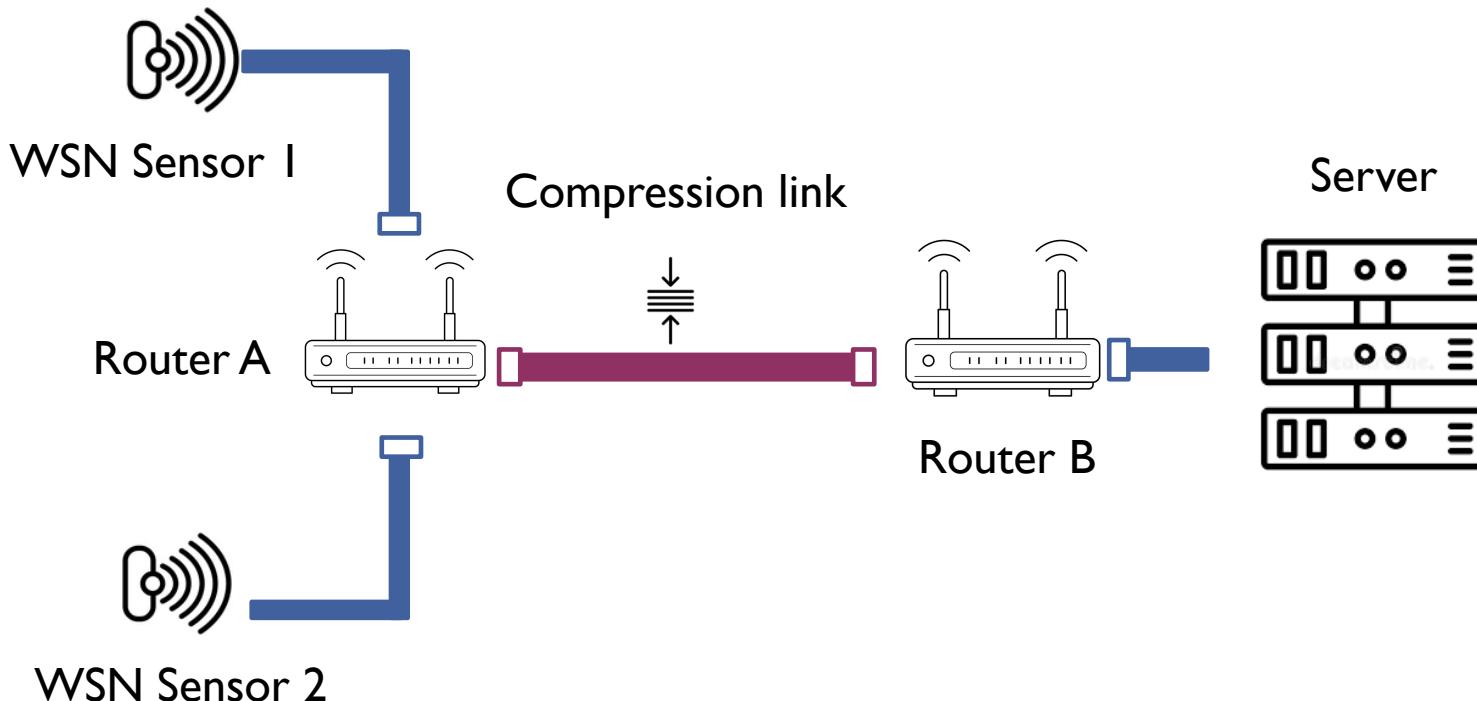
More and more industries adapt to WSNs and generate WSN node in the network. Loss-less data compression is essential solution to handle the high data traffic produced by WSNs. Hence, simulating loss-less compression link in open-source Network Simulator 3 (NS3) and validate the simulated environment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

Overview



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.

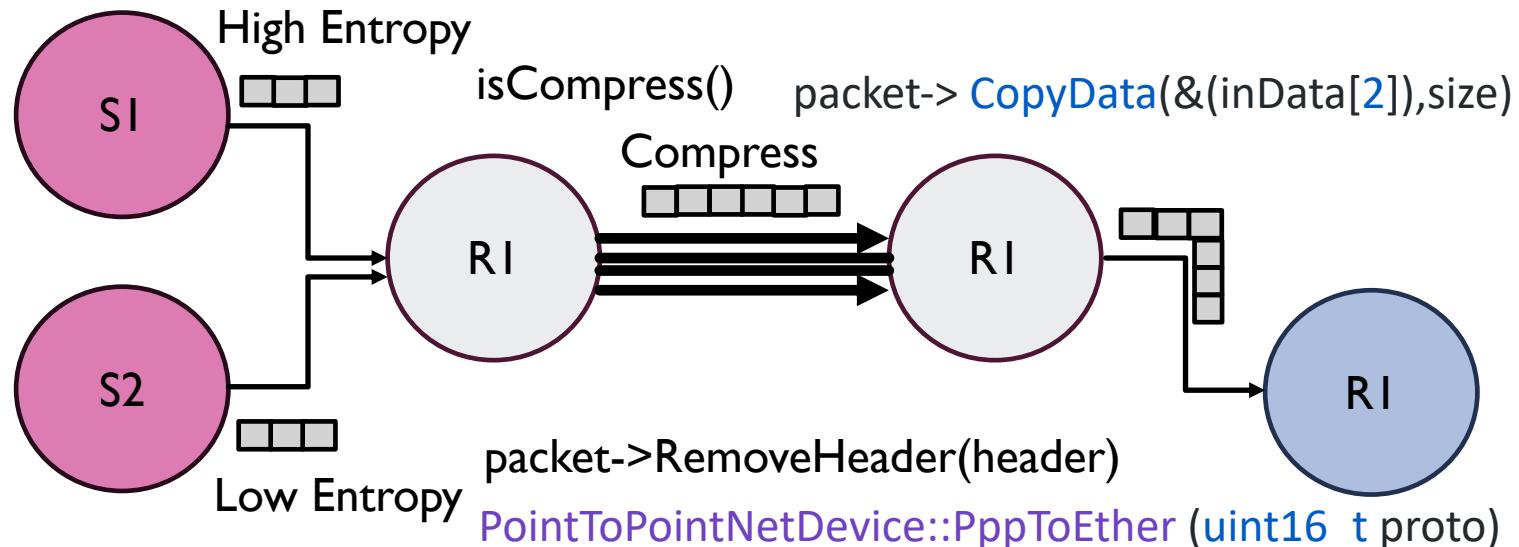
LZS Compression

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

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

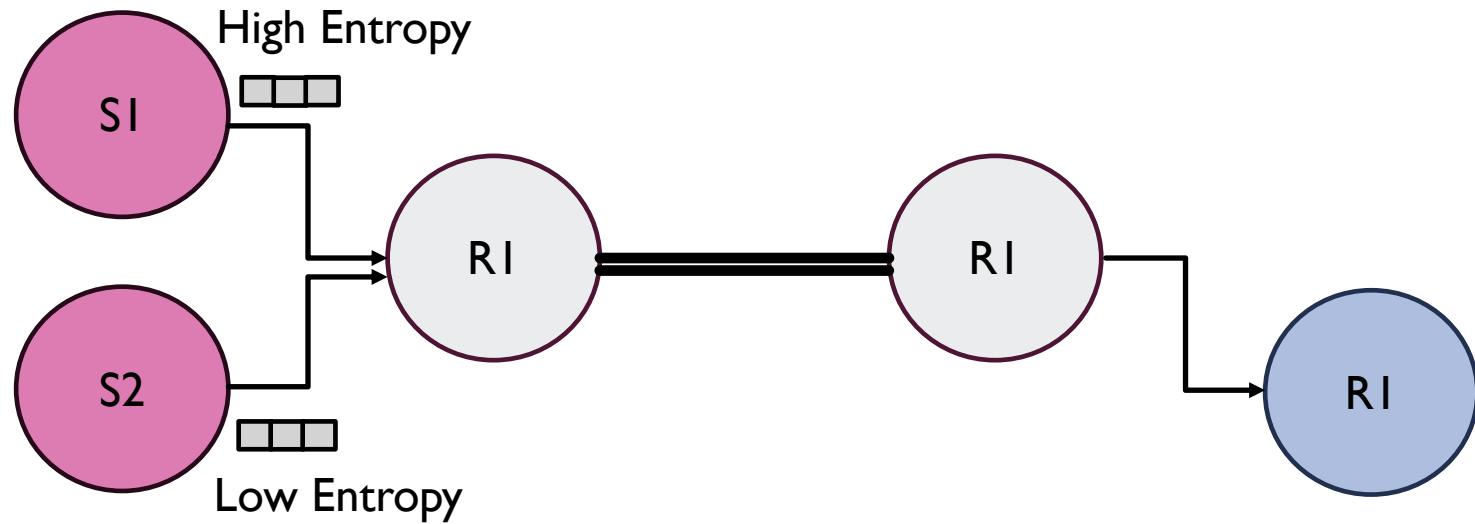
System Architecture

PointToPointNetDevice::Send()

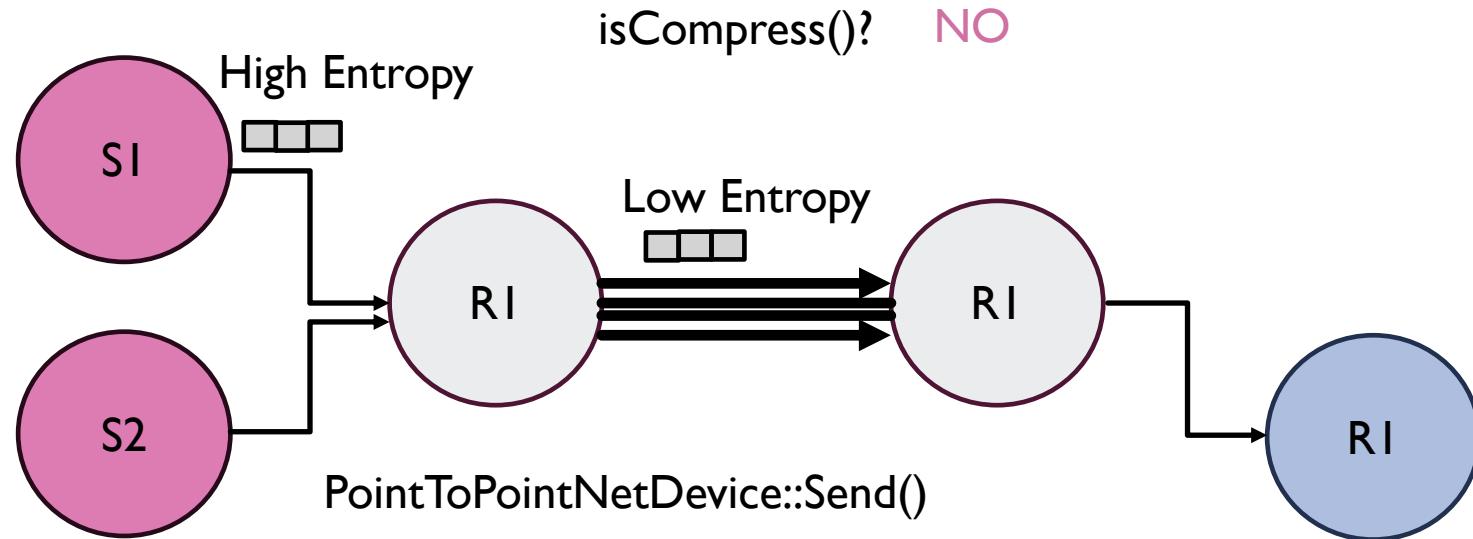


PointToPointNetDevice::Send()

System Architecture



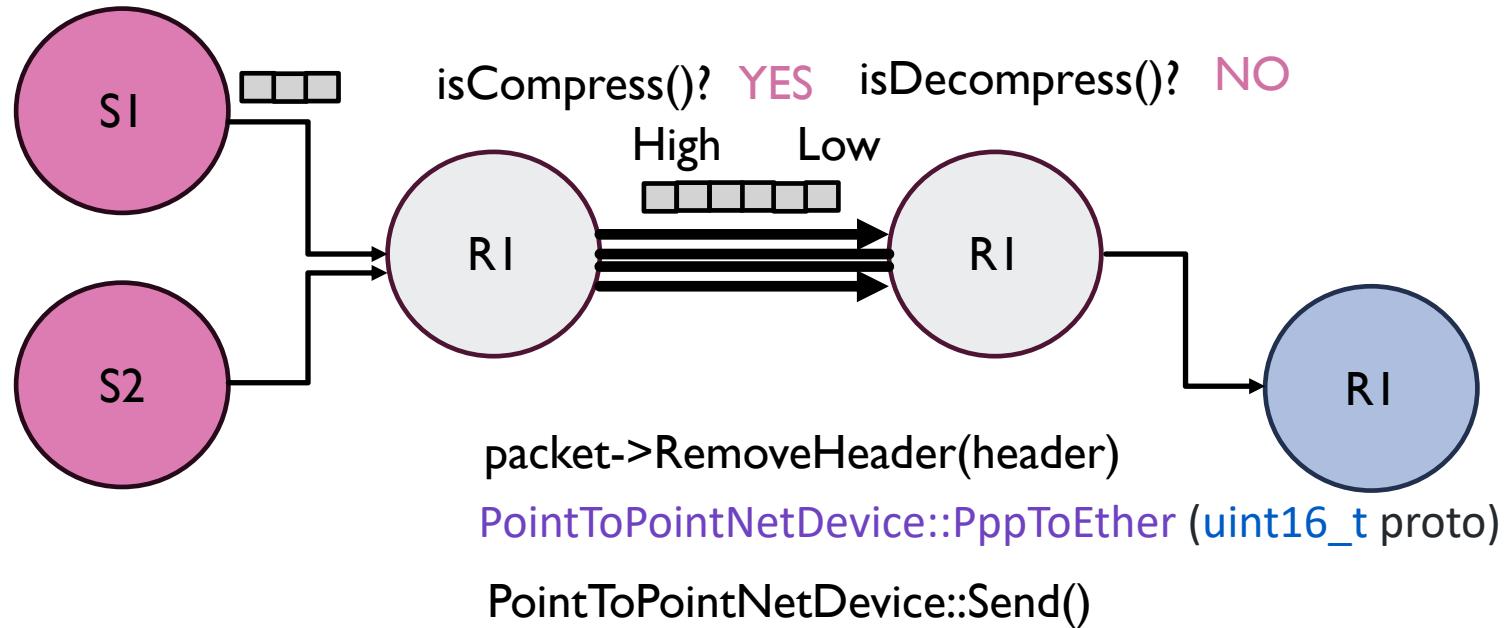
System Architecture



System Architecture

PointToPointNetDevice::Send()

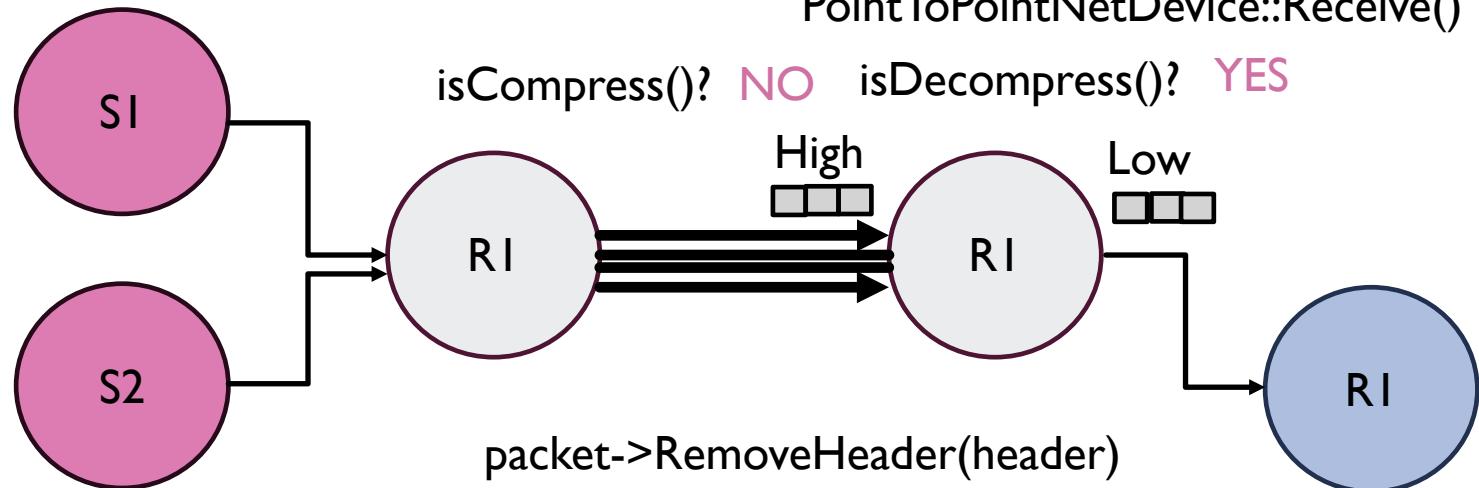
packet-> **CopyData**(&(inData[**2**]),size)



System Architecture

PointToPointNetDevice::Send()

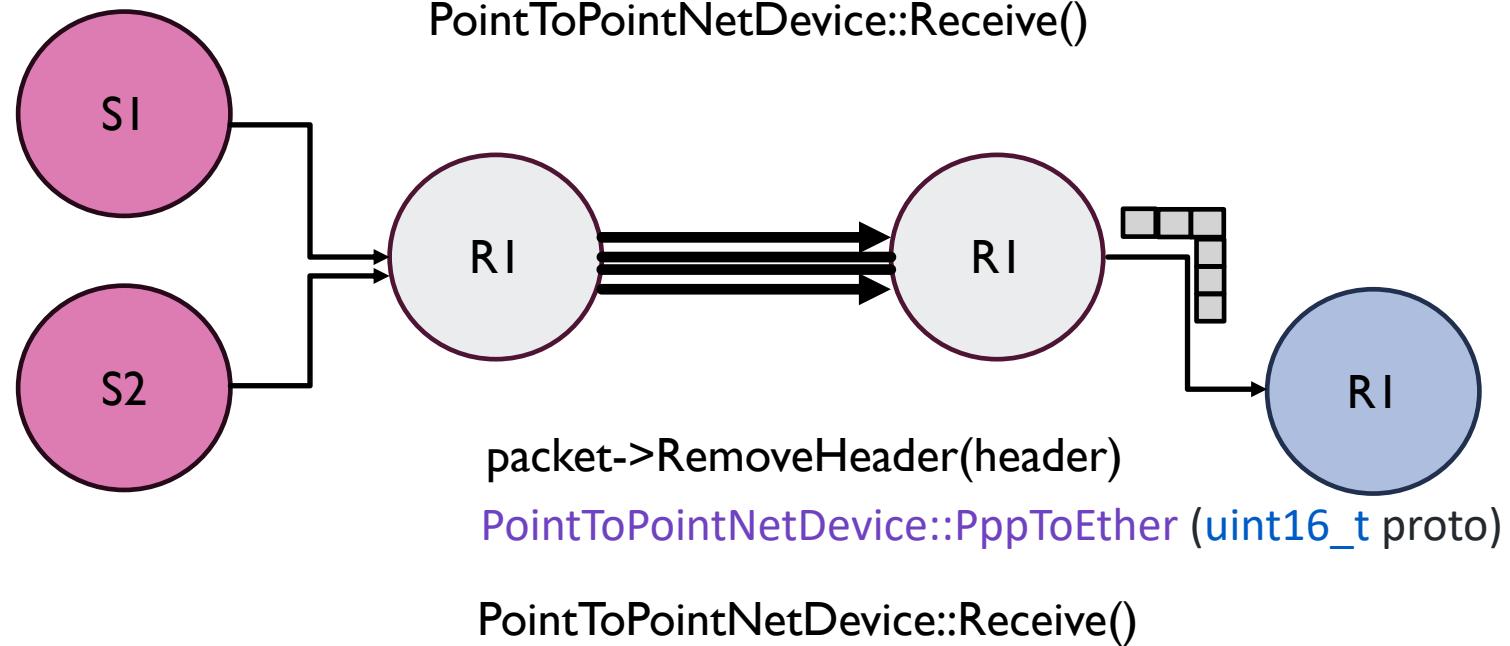
packet-> **CopyData**(&(inData[**2**]),size)



packet->RemoveHeader(header)

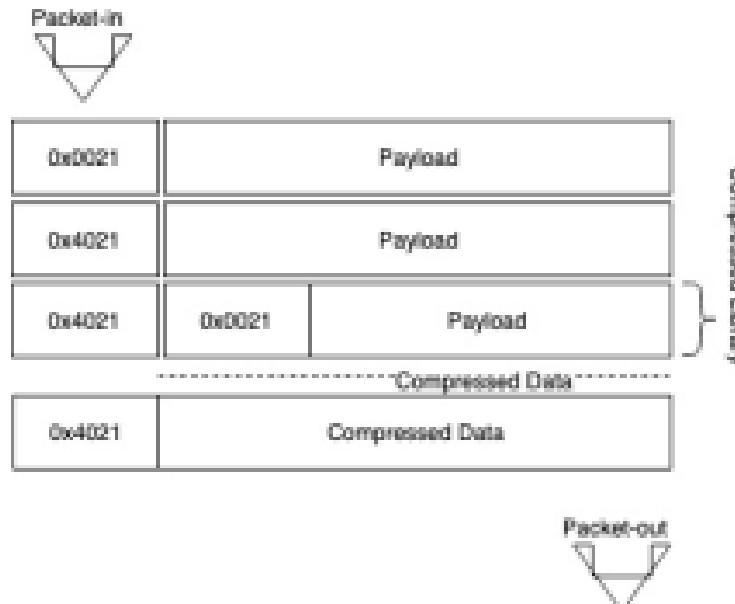
PointToPointNetDevice::PppToEther (**uint16_t** proto)

System Architecture



System Architecture

Packet arrival



Packet Departure

System Design

NS3:

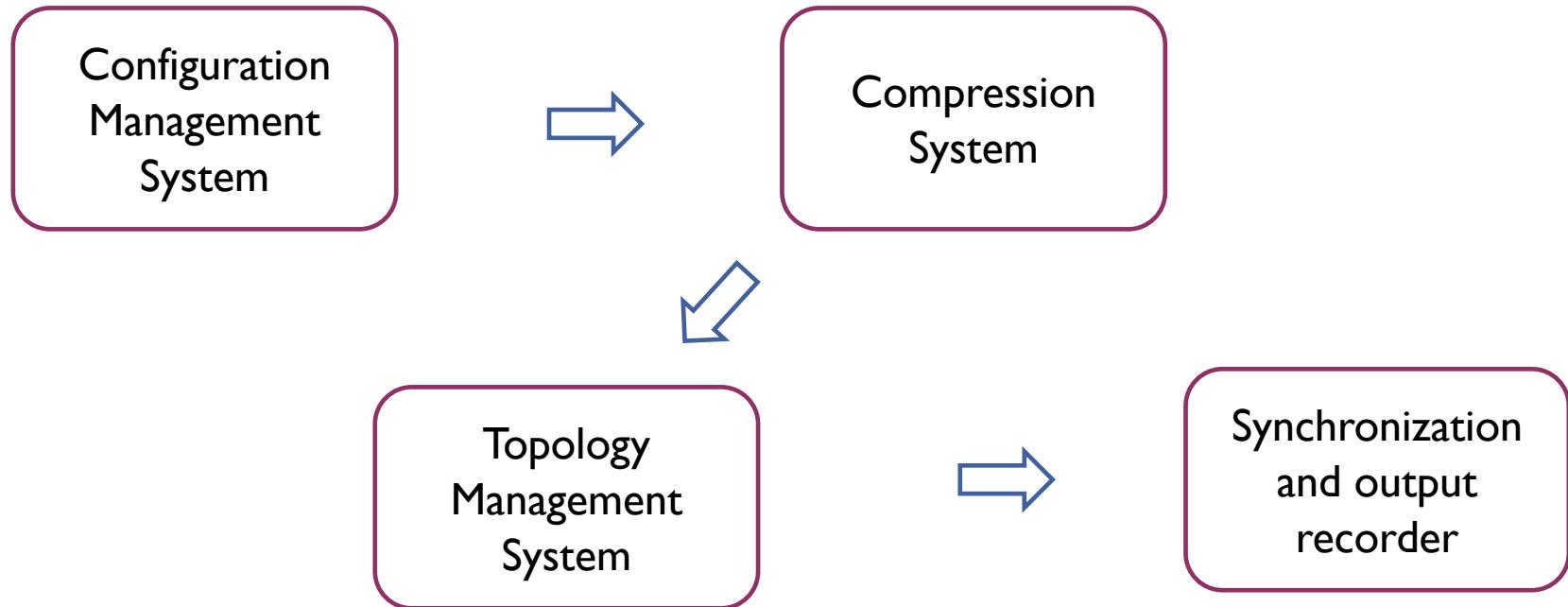
- .udp-client.cc
- .udp-server.cc
- .point-to-point-net-device.cc
- .point-to-point.cc (application)

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

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

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

System Design

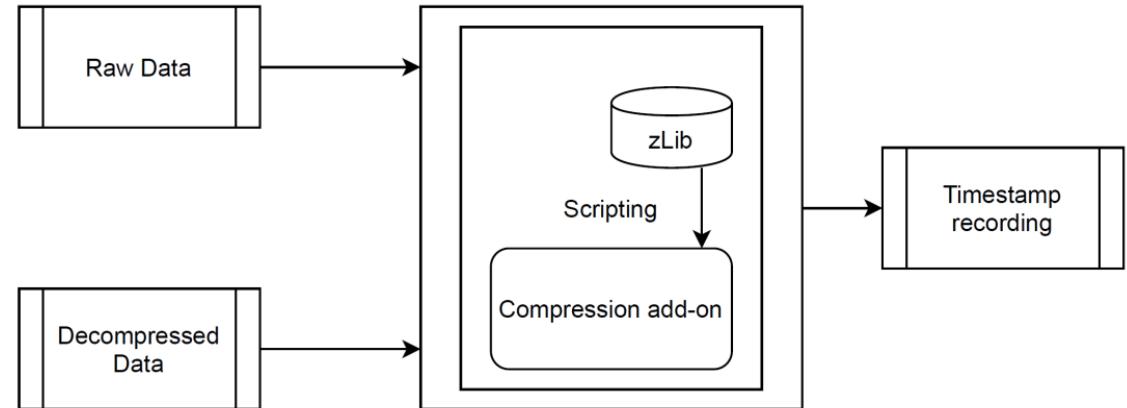


Detection Tool

$$\Delta tH_{\text{HighEntropy}} = T_{\text{FirstPktArrival}} - T_{\text{LastPktArrival}}$$

$$\Delta tL_{\text{LowEntropy}} = T_{\text{FirstPktArrival}} - T_{\text{LastPktArrival}}$$

Detection Factor = $\Delta tH - \Delta tL$



Build the Feature

```
/*
 * 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

29 9.384669	192.168.250.102	192.168.250.106	TCP	66 5001 > 40519 [ACK] Seq=1 Ack=7265 Win=65560 Len=0 Tsvl=278694 Tsecr=3598220
30 9.384693	192.168.250.107	192.168.250.107	TCP	1514 40519 > 5001 [ACK] Seq=20294 Ack=1 Win=14608 Len=1448 Tsvl=3598221 Tsecr=278694
31 9.385060	192.168.250.102	192.168.250.102	TCP	1514 40519 > 5001 [ACK] Seq=21745 Ack=1 Win=14608 Len=1448 Tsvl=3598221 Tsecr=278694
32 9.385101	192.168.250.102	192.168.250.102	TCP	1514 40519 > 5001 [ACK] Seq=23114 Ack=1 Win=14608 Len=1448 Tsvl=3598222 Tsecr=278694
33 9.387243	192.168.250.106	192.168.250.106	TCP	66 5001 > 40519 [ACK] Seq=1 Ack=10161 Win=65560 Len=0 Tsvl=278694 Tsecr=3598221
34 9.387265	192.168.250.106	192.168.250.106	TCP	1514 40519 > 5001 [ACK] Seq=20294 Ack=1 Win=14608 Len=1448 Tsvl=3598221 Tsecr=278694
35 9.387406	192.168.250.102	192.168.250.102	TCP	1514 40519 > 5001 [ACK] Seq=26089 Ack=1 Win=14608 Len=1448 Tsvl=3598222 Tsecr=278694
36 9.387450	192.168.250.102	192.168.250.102	TCP	66 5001 > 40519 [ACK] Seq=1 Ack=13057 Win=14608 Len=1448 Tsvl=3598222 Tsecr=278694
37 9.389307	192.168.250.102	192.168.250.102	TCP	1514 40519 > 5001 [ACK] Seq=27537 Ack=1 Win=14608 Len=1448 Tsvl=3598222 Tsecr=278694

Frame 30: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)

WTAP_ENCAPS =

Arrival Time: Aug 28, 2012 14:31:29.699756000 AUS Eastern Standard Time

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1341282829.699756000 seconds

[Time delta from previous captured frame: 0.000299000 seconds]

[Time delta from previous displayed frame: 0.000299000 seconds]

[Time since reference or first frame: 9.384968000 seconds]

Frame Number: 30

Frame Length: 1514 bytes (12112 bits)

Capture Length: 1514 bytes (12112 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ip:tcp:data]

[Coloring Rule Name: TCP]

[Coloring Rule Priority: 100]

▀ Ethernet II Src: Gemtekte_8cae:6d (00:21:00:8c:ae:6d), Dst: LiteonTe_id:18:b0 (68:a3:c4:1d:18:b0)

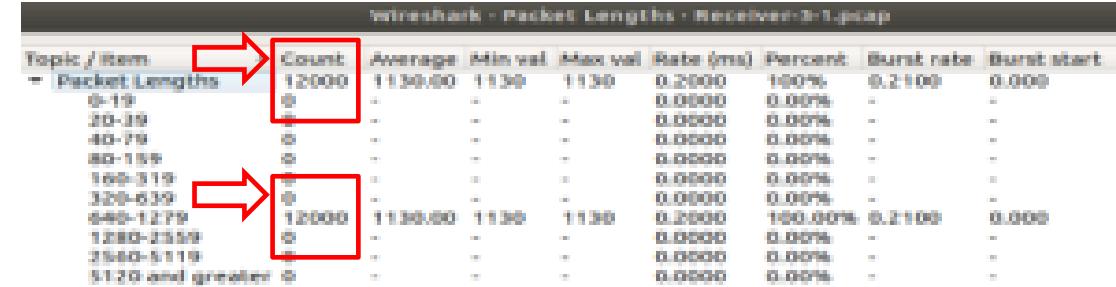
▀ Internet Protocol Version 4, Src: 192.168.250.106 (192.168.250.106), Dst: 192.168.250.102 (192.168.250.102)

▀ Transmission Control Protocol, Src Port: 40519 (40519), Dst Port: 5001 (5001), Seq: 20297, Ack: 1, Len: 1448

▀ Data (1448 bytes)

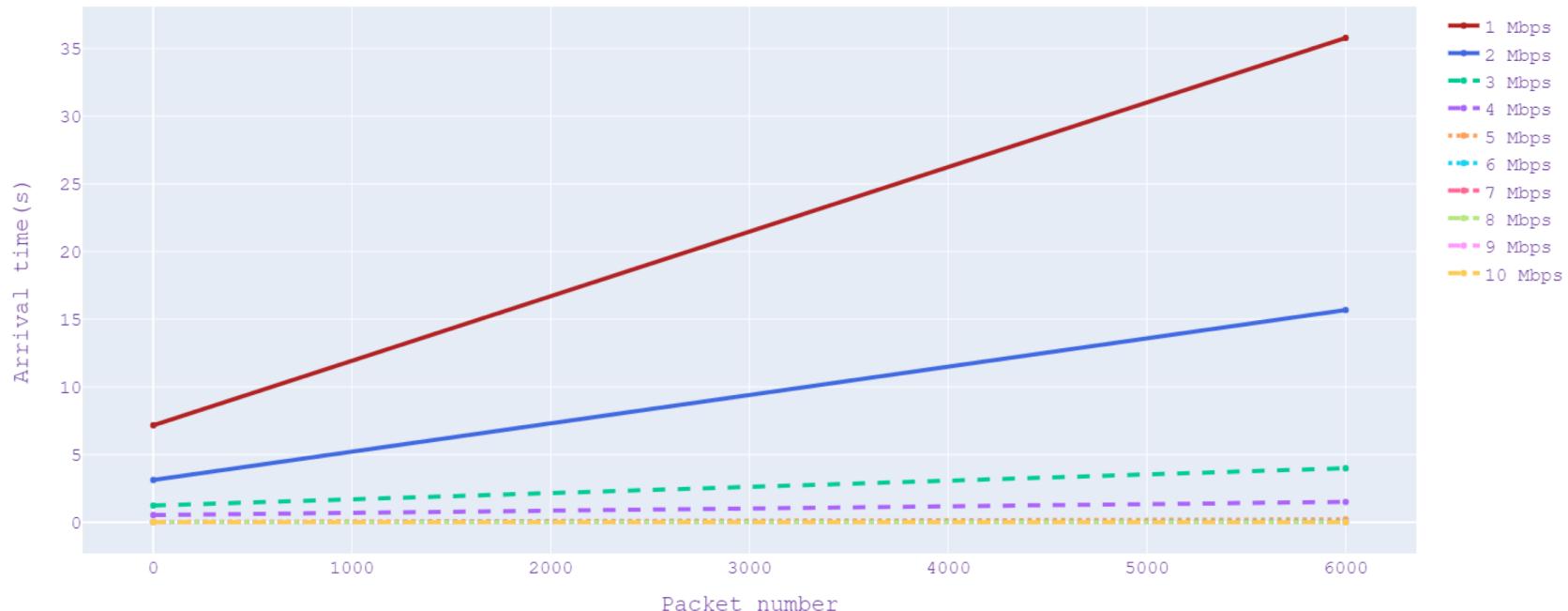
0000	05 82 c4 1d 18 b0 00 21 00 1c ae 6d f8 00 45 00	h... .l ... m..E
0001	05 dc 38 bc 40 00 40 06 86 c0 a8 Fa 00 c0 a8	h... .l ... m..E
0002	f4 66 9e 47 13 89 90 f4 2f 19 1c 33 41 48 80 10	f...G... /...3AH..
0003	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0...0...0...0...0...0...0...0...0...0...0...0...0...0...0...0
0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0.234567 89012345

Wireshark Packet Analyzer

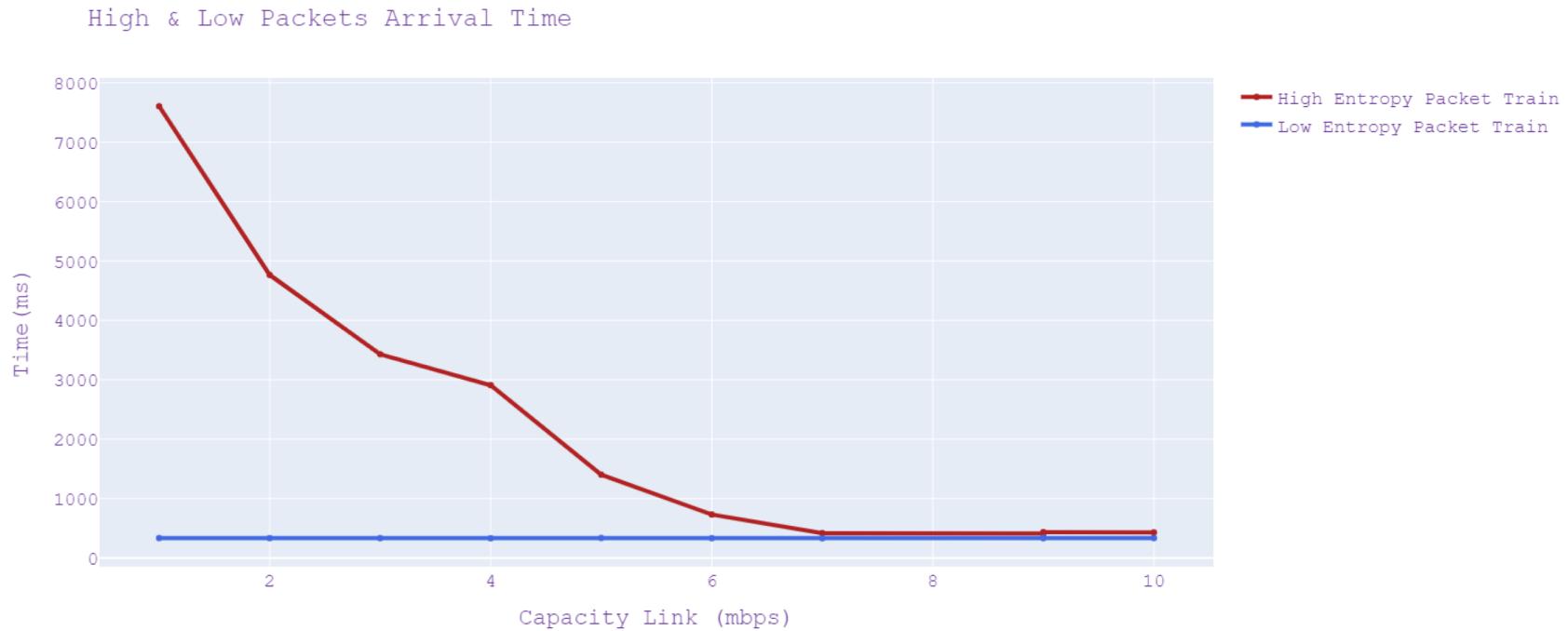


validation

Compression in capacity link bottle-neck



validation



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

- U.S. National Science Foundation (NSF), “A discrete-event network simulator for internet systems, NS-3 Network Simulator”,
<https://www.nsnam.org/>, 2011-2021.
- V. Pournaghshband, A. Afanasyev, P. Reiher, “End-to-end detection of compression of traffic flows by intermediaries”, 2014 IEEE Network Operations and Management Symposium (NOMS), May 2014, DOI: 10.1109/NOMS.2014.6838247 , pp.1–8.
- J. Azar, A. AMakhoul, R. Couturier, J. Demerjian, “Robust IoT time series classification with data compression and deep learning”, Neurocomputing, Vol. 398, 2011-2021, DOI: 10.1016/j.neucom.2020.02.097, 2020, Elsevier, pp. 222–234.