



# Wireshark Dissector

Chavoosh Ghasemi & John Dellaverson & Davide Pesavento



## Goals at the Outset:

- Make TLV-TYPE decoding context-aware (#4185, #4518)
- Improve info displayed on unrecognized and out-of-order TLV elements (#3197)
- Produce sample pcap traces for above cases
- Compare before/after



## What's the upside?

- Wireshark is a handy, popular tool for developer
- Useful tool for NDN packet tracing and network debugging



# Context-Aware TLV-TYPE decoding

## Problem

Wireshark interprets each element independent of other elements

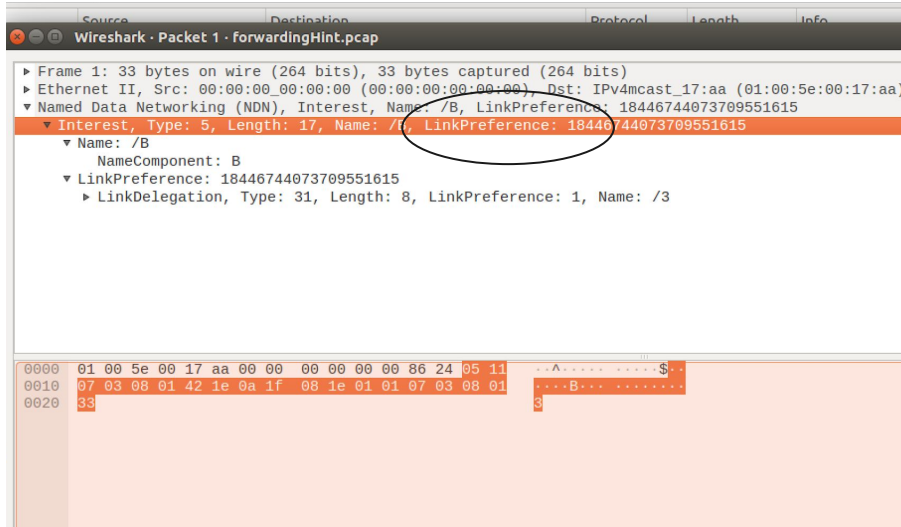
- Wireshark dissector does not recognize two TLV types with the same assigned number (redmine #4185)
  - 0x1E is **ForwardingHint** under Interest but **Preference** under Delegation
- Name Components should not interpret as elements denoted by TLV-TYPE
  - E.g., TLV-TYPE 0x21 normally means CanBePrefix but within a name is SegmentNameComponent



# Pcap Files

- Engineered to distinguish **Forwarding Hint** from **Preference TLV-TYPE**
- Engineered not to interpret Name Components as elements denoted by TLV-TYPE

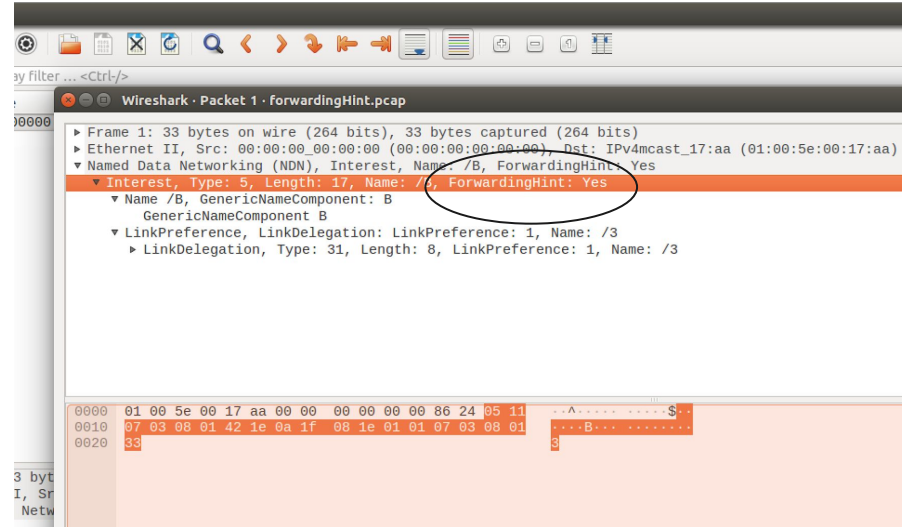
# Before / After



Wireshark - Packet 1 - forwardingHint.pcap

- ▶ Frame 1: 33 bytes on wire (264 bits), 33 bytes captured (264 bits)
- ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: IPv4mcast\_17:aa (01:00:5e:00:17:aa)
- ▼ Named Data Networking (NDN), Interest, Name: /B, LinkPreference: 18446744073709551615
  - ▼ Interest, Type: 5, Length: 17, Name: /B, LinkPreference: 18446744073709551615
    - ▼ Name: /B
      - NameComponent: B
      - ▼ LinkPreference: 18446744073709551615
        - ▶ LinkDelegation, Type: 31, Length: 8, LinkPreference: 1, Name: /3

0000 01 00 5e 00 17 aa 00 00 00 00 00 86 24 05 11 ...A.....\$.  
0010 07 03 08 01 42 1e 0a 1f 08 1e 01 01 07 03 08 01 ...B.....  
0020 33 3



Wireshark - Packet 1 - forwardingHint.pcap

- ▶ Frame 1: 33 bytes on wire (264 bits), 33 bytes captured (264 bits)
- ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: IPv4mcast\_17:aa (01:00:5e:00:17:aa)
- ▼ Named Data Networking (NDN), Interest, Name: /B, ForwardingHint: Yes
  - ▼ Interest, Type: 5, Length: 17, Name: /B, ForwardingHint: Yes
    - ▼ Name /B, GenericNameComponent: B
      - GenericNameComponent B
      - ▼ LinkPreference, LinkDelegation: LinkPreference: 1, Name: /3
        - ▶ LinkDelegation, Type: 31, Length: 8, LinkPreference: 1, Name: /3

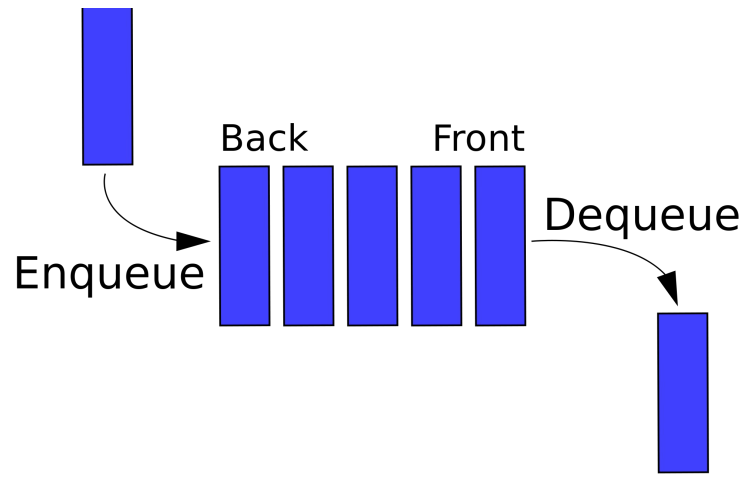
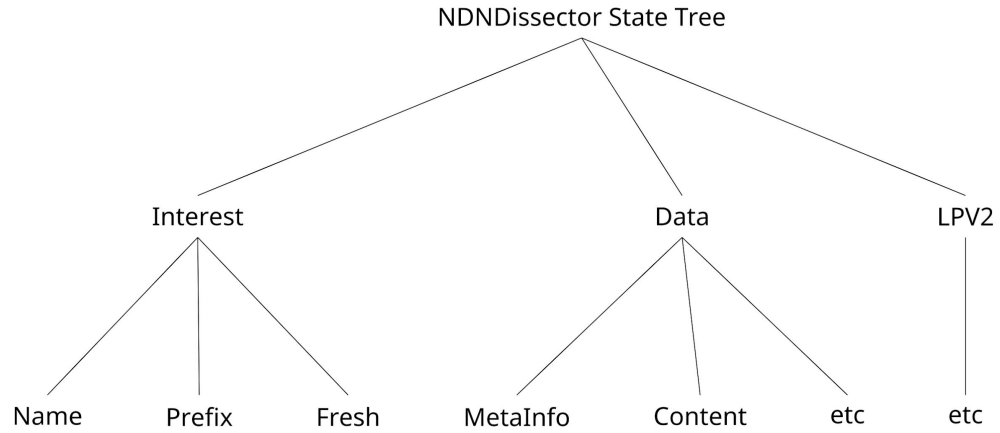
0000 01 00 5e 00 17 aa 00 00 00 00 00 86 24 05 11 ...A.....\$.  
0010 07 03 08 01 42 1e 0a 1f 08 1e 01 01 07 03 08 01 ...B.....  
0020 33 3

33 bytes  
I, Sr  
Netw



# What could be next?

- Currently, we process blocks in a queue
  - Child not strictly connected to parent.
- Sol'n: Process in strict order (DFS on the tree).
  - Resolves the issue.
  - Also makes issue 4 (processing out-of-order elements) easier







# Question