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

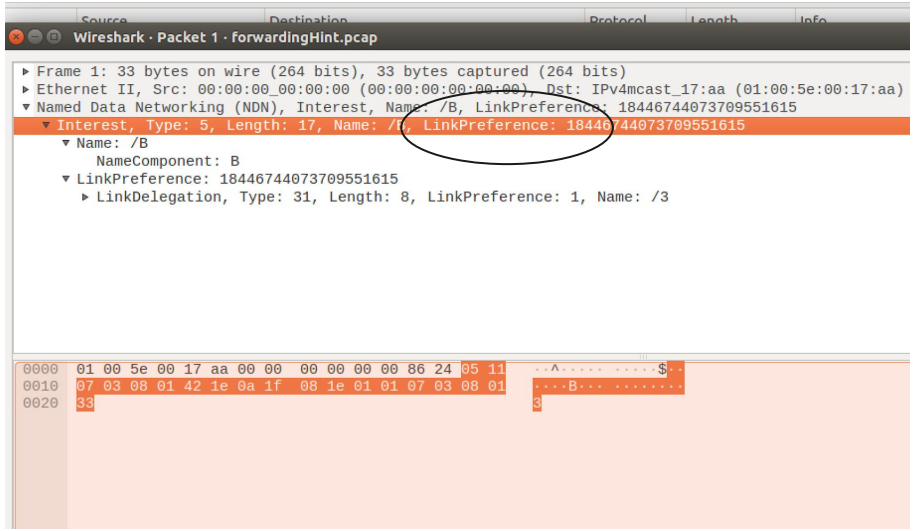
Wireshark interprets each element independent of other elements

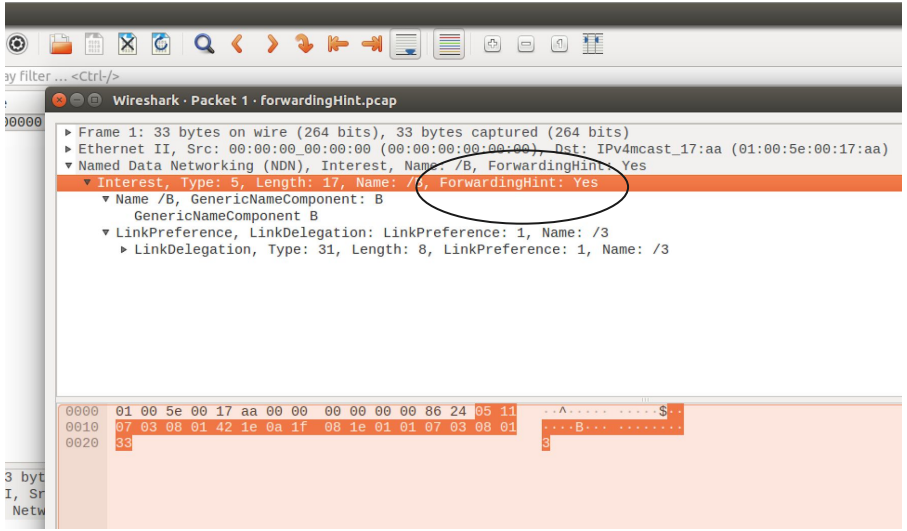
- Wireshark dissector does not recognize two TLV types with the same assigned number (redmine #4185)
  - 0x1E is **ForwardingHint** under Interest but **Preference** under Delegation
- Name Components should not interpret as elements denoted by TLV-TYPE
  - E.g., TLV-TYPE 0x21 normally means CanBePrefix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

Pcap Files

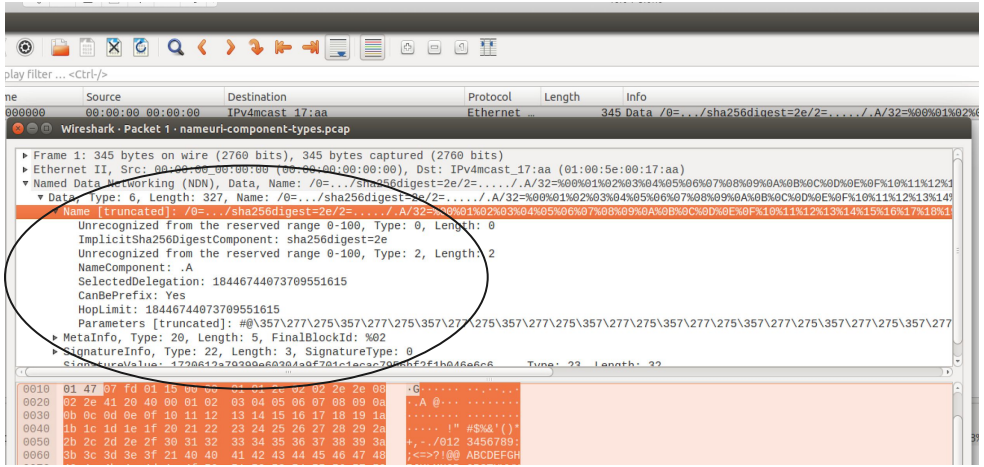
- Engineered to distinguish **Forwarding Hint** from **Preference** TLV-TYPE
- Engineered not to interpret Name Components as elements denoted by TLV-TYPE

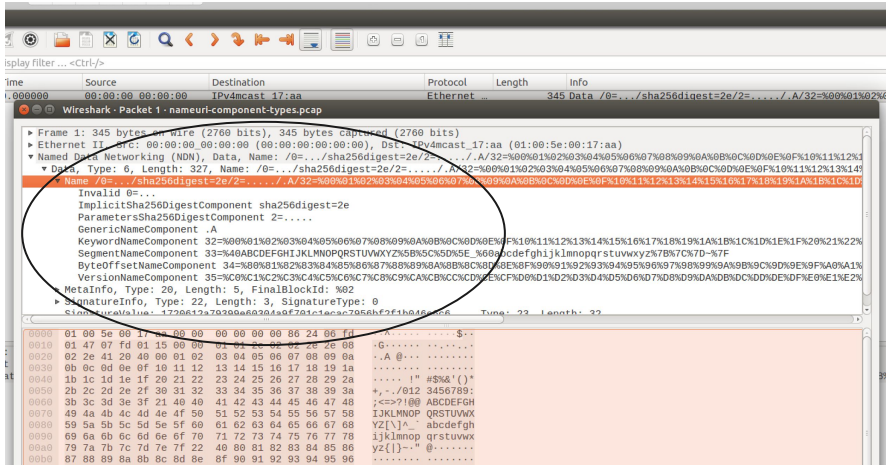
Before / After





Before / After

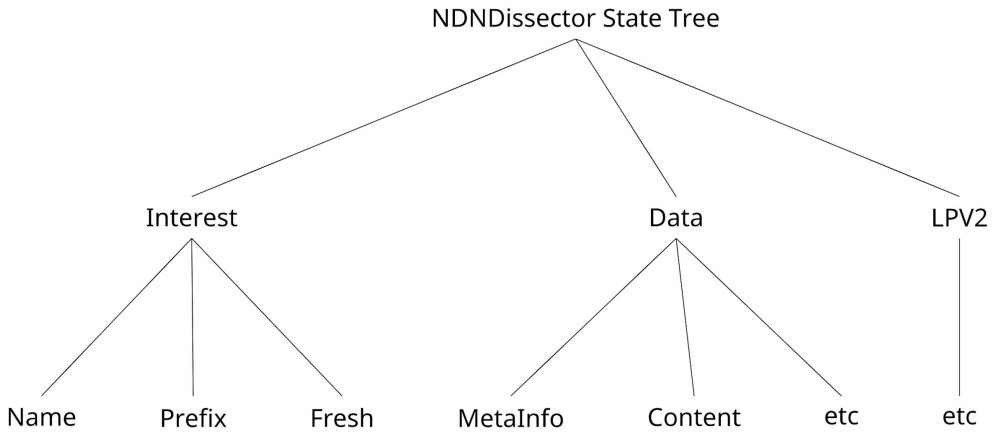


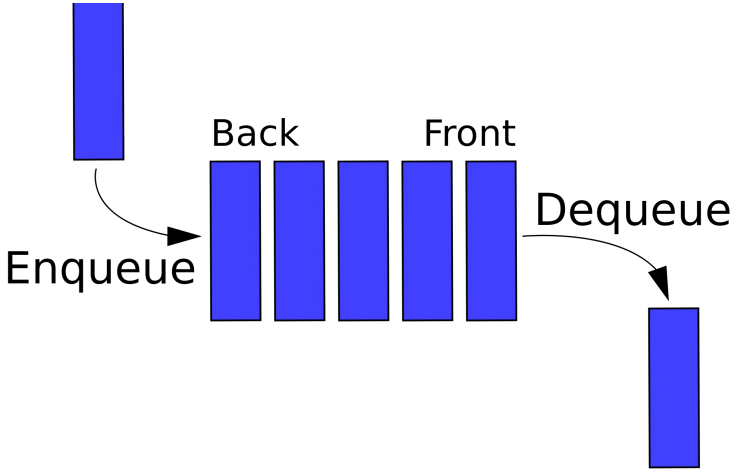


What could be next?

- 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