DTN tunneling

Christos-Alexandros Sarros
Research assistant,
Athena Research and Innovation Center
Problem statement

• NDN originally developed for low-delay, highly-connected network environments

• Intermittent connectivity: no support out-of-the-box

• Additional mechanisms are needed

• Breadcrumbs routing limitations
Breadcrumbs 1:
End-to-end connectivity

- Interest
- Data
Breadcrumbs 2: No end-to-end connectivity

Interest

Data
Breadcrumbs 3:
No end-to-end connectivity

Interest

Data

?
Opportunistic mobile networks

Interest

Data
Opportunistic mobile networks

Interest

Data
Problem statement

**Goal:** Enhance NDN to provide delay intermittent connectivity support

**Solution 1:**
- Try to build delay/disruption tolerant functionality into NDN
- Totally possible

**Example:**
- Data is lost when transferred back. Consumer has to send interest again, but not efficient (very large delay until next contact).
- One solution would be to have the data mule retransmit the Interest packet, assuming responsibility for content retrieval.
- Just arrived at some form of the DTN custody transfer mechanism
- Could perform Interest flooding, polling or both – inefficiency problems again
- Forward Interests only to nodes most probable to deliver them/return the Data
- After a while we will start mimicking DTN routing algorithms
- Accept unsolicited Data: altering NDN core communication primitives, possible security problems
Solution 2: Integrating DTN

- **First thoughts:** DTN an already well-mature technology, with several existing implementations

- **Idea:** Instead of implementing delay/disruption-tolerant functionality from scratch in NDN, leverage the existing implementations

- **Solution:** DTN as an NDN underlay (or vice-versa)

- **Practically:** Create a DTN face in NFD, in order to tunnel NDN packets through DTN islands (or isolated data mules)
DTN can... (1)

(a) Enable opportunistic forwarding between ICN nodes
DTN can... (2)

(b) Accommodate delays
DTN can... (3)

(c) Increase reliability
Approach tradeoffs

• **On the plus side:**
  • Compatibility with the original NDN architecture
  • Facilitates NDN deployment over existing DTN implementations
  • Intermittent connectivity handling is abstracted from NDN

• **On the minus side:**
  • Not a native mechanism; NDN not delay tolerant by itself
  • Extra layers = overhead
  • Not a single extra layer: The bundle protocol also an overlay
Protocol stack
Implementation

- Linux/Android
- Used the IBR-DTN Bundle Protocol implementation as DTN transport
- Software stack:
Android implementation

• A closer look
Demos

1. Vid: alternate mobile returning Data

2. DTN tunneling for service deployment in remote locations (demo session)

- Code available @ https://github.com/umobileproject
Questions?

• Thank you :}

Thank you :)
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124