

DTN2 GO

DELAY/DISRUPTION-

**TOLERANT
NETWORKING**

Motivation

Situations without a reliable uplink

- ▶ Internet access is blocked
- ▶ Destroyed telecommunication infrastructure
- ▶ Transmissions from rural areas
- ▶ Deep space communications

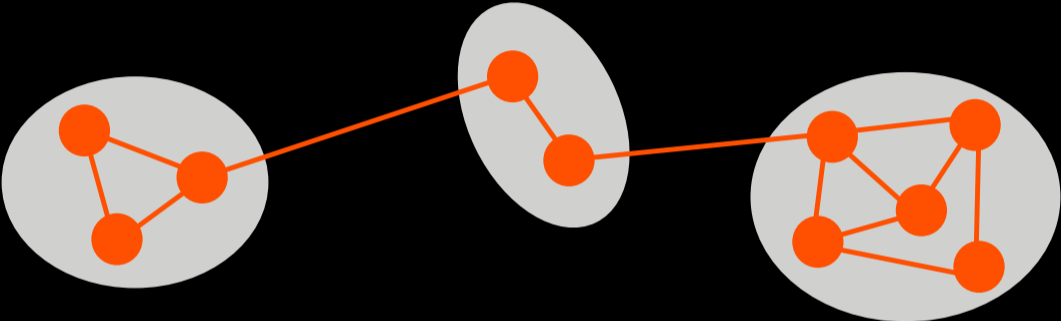
Motivation

Situations without a reliable uplink

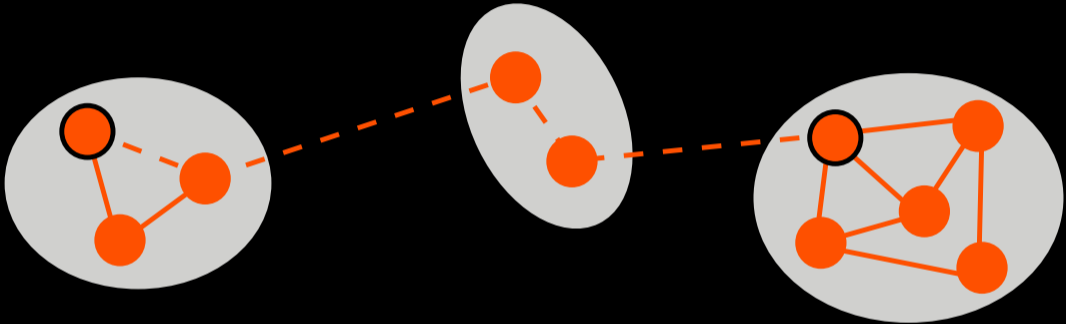
- ▶ Internet access is blocked
- ▶ Destroyed telecommunication infrastructure
- ▶ Transmissions from rural areas
- ▶ Deep space communications

Typical solution: Wireless mesh networking

Your regular mesh network

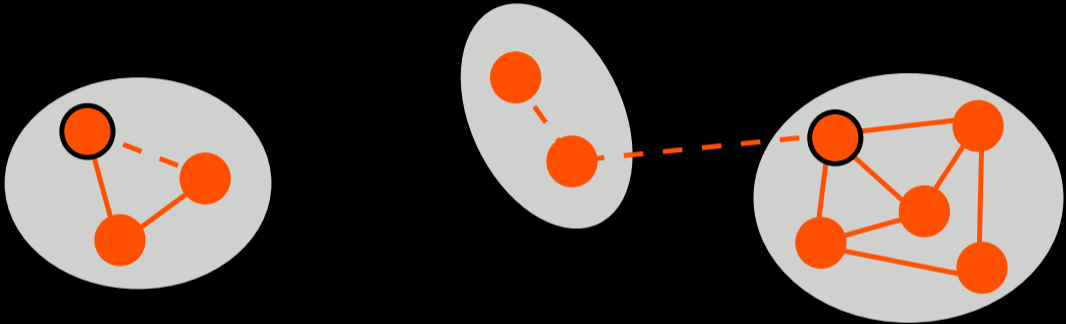


Your regular mesh network



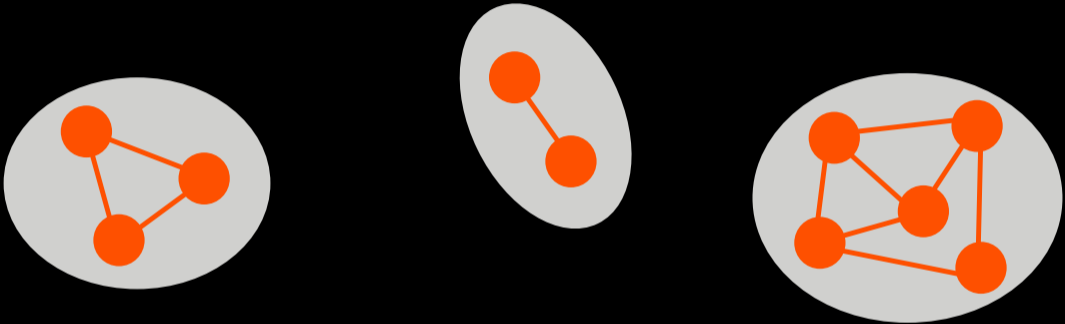
- ▶ TCP requires a continuous connection for handshake and ACKing segments

Your regular mesh network



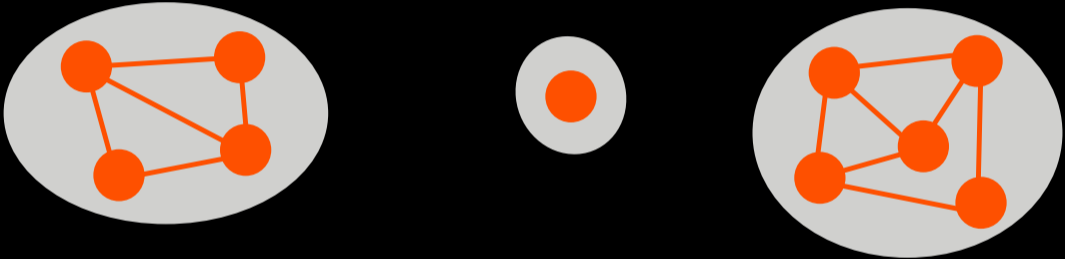
- ▶ TCP requires a continuous connection for handshake and ACKing segments
- ▶ TCP does not work on partitioned networks

Your regular mesh network



- ▶ TCP requires a continuous connection for handshake and ACKing segments
- ▶ TCP does not work on partitioned networks
- ▶ IRL, components are often isolated

Your regular mesh network



- ▶ TCP requires a continuous connection for handshake and ACKing segments
- ▶ TCP does not work on partitioned networks
- ▶ IRL, components are often isolated and nodes are switching between them

Technical requirements

- ▶ No end-to-end connections
- ▶ Reduce or eliminate round-trips
- ▶ Ensure functionality for disconnected networks
- ▶ Taking node mobility into account

The DTN approach

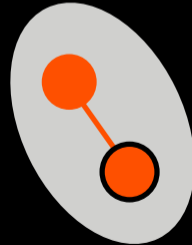
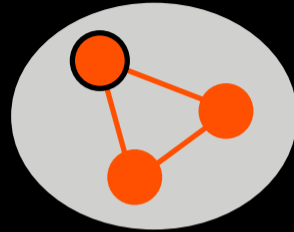
- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled

DTN: Delay / Disruption-Tolerant Networking

The DTN approach

- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled

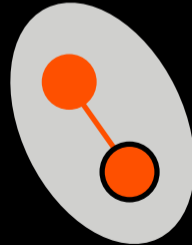
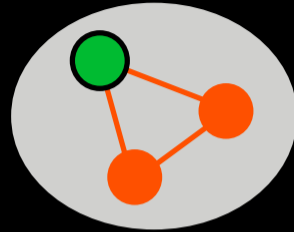
DTN: Delay / Disruption-Tolerant Networking



The DTN approach

- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled

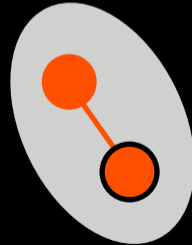
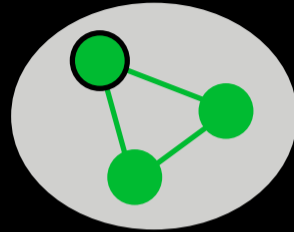
DTN: Delay / Disruption-Tolerant Networking



The DTN approach

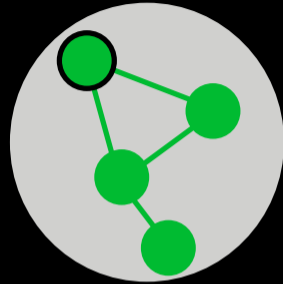
- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled

DTN: Delay / Disruption-Tolerant Networking



The DTN approach

- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled



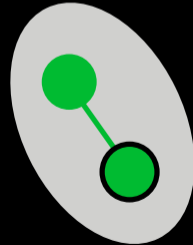
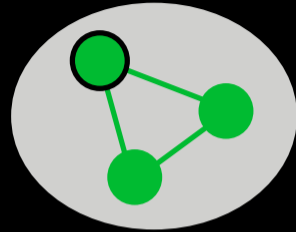
DTN: Delay / Disruption-Tolerant Networking



The DTN approach

- ▶ Packets are transmitted hop-to-hop in a store-carry-forward manner
- ▶ Nodes redistribute packets on contact
- ▶ Next contact might be created opportunistically or scheduled

DTN: Delay / Disruption-Tolerant Networking



Bundle Protocol Version 7

- ▶ Describes both a DTN architecture and protocol
- ▶ Protocol draft `draft-ietf-dtn-bpbis-17`
- ▶ Aims to obsolete Bundle Protocol Version 6, RFC 5050

Bundle

Bundle

Primary Block

Version: 7
Control Flags:
 Status requested for reception
CRC Type: *CRC32*
Destination EID: *dtm://dst/*
Source node EID: *dtm://src/*
Report-to EID: *dtm://src/*
Creation Timestamp: (0, 23)
Lifetime: *12h*
CRC Value: 67 75 6D 6F

Hop Count Block

Type Code: 10
Number: 2
Control Flags: *None*
CRC Type: *CRC32*
Data: (64, 42)
CRC Value: F9 85 2A 5D

Payload Block

Type Code: 1
Number: 1
Control Flags: *None*
CRC Type: *CRC32*
Data: "Hello 36C3"
CRC Value: 6E 18 1A C1

Canonical Blocks

dtm7-go

- ▶ Software implementation for a DTN based on the BP
- ▶ Both DTN router and application interface
- ▶ Usable as a Golang library

Bundle Exchange

- ▶ Convergence Layer
 - ▶ Transport technology for Bundles between nodes
 - ▶ Implemented: TCP, LoRaPHY
 - ▶ Possible: Bluetooth, Email, QR code, Pigeon, ...
- ▶ Routing
 - ▶ Selection of neighbors for Bundle delivery
 - ▶ Implemented: DTLSR, Epidemic Routing, PRoPHET, Spray and Wait

Using dtn7-go

Bundles can be sent by using

- ▶ RESTful API
- ▶ Including dtn7-go as a library

Delay/Disruption-Tolerant Networking with dtn7-go

<https://github.com/dtn7/dtn7-go>

<https://dtn7.github.io/>