Postellation: an Enhanced Delay-Tolerant Network (DTN) Implementation with Video Streaming and Automated Network Attachment

Marc Blanchet, Simon Perreault, Jean-Philippe Dionne

Viagénie

Marc.Blanchet@viagenie.ca

http://viagenie.ca

Plan

VIAGĒNIE

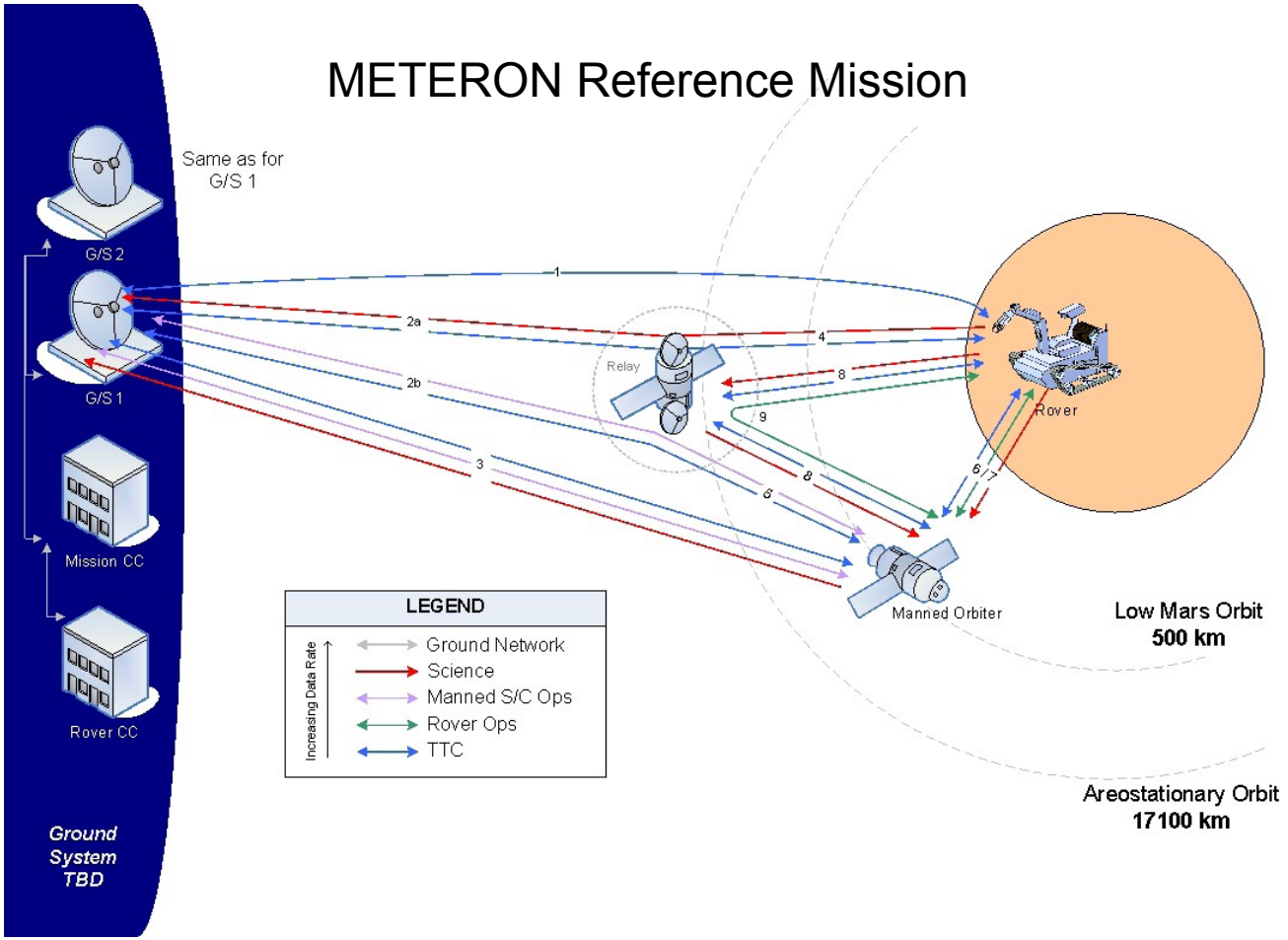
- Background
- Key Design Considerations
- Features
- HTTP over DTN
- DTN News Service
- Virtual DTN Cloud and demo

POSTELLATION

Delay-Tolerant Networks

- Requirements:
  - Delay-tolerant
  - Disruption-tolerant
  - Network
    - instead of point to point links
  - Reliability
- Basic atomic element is a bundle (RFC5050)
- Carried over a convergence layer:
  - Terrestrial: TCP, UDP
  - Space: Licklider Transport Protocol (RFC5326)
    - over CCSDS links
- Store and Forward

Example of DTN



- A Rover Control Center which manages the overall robotic operations
- A Mission Control Center which manages the space mission by transmitting Telemetry/Command messages via the Ground Stations
- A Manned Orbiter (or Surface Habitat) from which crew teleoperate the surface robotic elements
- A Relay Satellite which interconnects:
- the Rovers with the Manned Orbiter/Surface Habitat, other surface Rovers, and; with the Ground

From: Multipurpose End-To-End Robotic Operations Network (METERON), ESA/NASA

Current Issues with DTN Software

- Some implementations are big, heavy, complicated
  - Many not ready for flying
- Applications to use DTN need to be written from scratch
  - No standardized API. Even standardized, new network API.
  - New logic
  - Application need to be deeply aware of DTN network layer
  - Consequence: long long development time. No reuse.
- Complicated usage for end-users
  - No current usage in terrestrial world.
  - Codepaths are not exercised.

Postellation

- Name comes from:
  - <post>ellation:
    - Postal service is store and forward “network”
    - Has optional “custody”
  - post<ellation>:
    - Constellation => network
- Project:
  - Implementation of DTN
  - DTN simulation cloud
- http://postellation.viagenie.ca

Key Design Considerations

- **Lean** Bundle protocol implementation
  - → good for embedded systems
- **Smart** HTTP proxy
  - → enabling Web/SOA application developers to use DTN "transparently"
  - → optimized video streaming
- **Easy** deployment of DTN networks
  - → enabling a much larger number of end-users to use DTN, develop a community, applications, ...

Features

- written in lean and “vanilla” C → for embedded systems
- Portable code: compiles/runs/tested on:
  - Linux (kernel 2.6+)
  - *BSD, MacOSX (Leopard, Snow Leopard)
  - Windows (from XP to W7)
  - RTEMS (4.10+)
- Bundle Protocol (RFC5050)
- Convergence Layers:
  - UDP, TCP and TCP-TLS
- Transport: IPv4 ***and IPv6***

Features (cont.)

- Included applications:
  - dtnping/dtnpong
  - dtnsend/dtnrecv
  - HTTP/HTTPS Proxy
  - RSS news service delivery, such as NASA news over DTN!
- Packagers for Windows, MacOSX and Linux
- Automated registration of nodes to our DTN node:
  - ***No configuration to do.***
  - ***And you are connected to the DTN network***

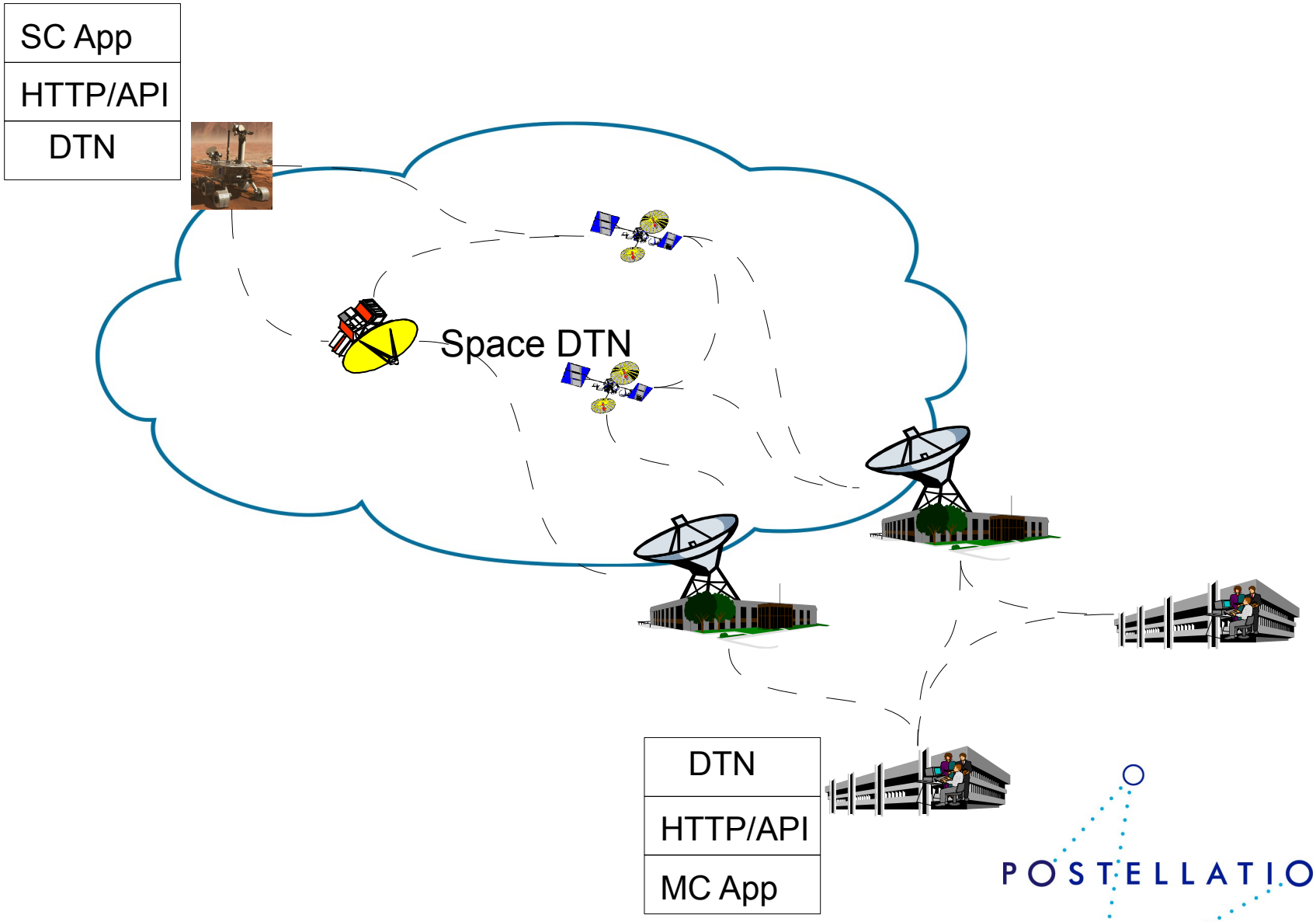
HTTP Proxy

- Support:
  - http
  - https
  - or any http tunnels
- Smarts to facilitate transparency of Web applications over DTN
- Implemented as a local proxy
  - For bundling HTTP requests into Bundles
- With a remote proxy
  - For unbundling HTTP requests and sending them over IP

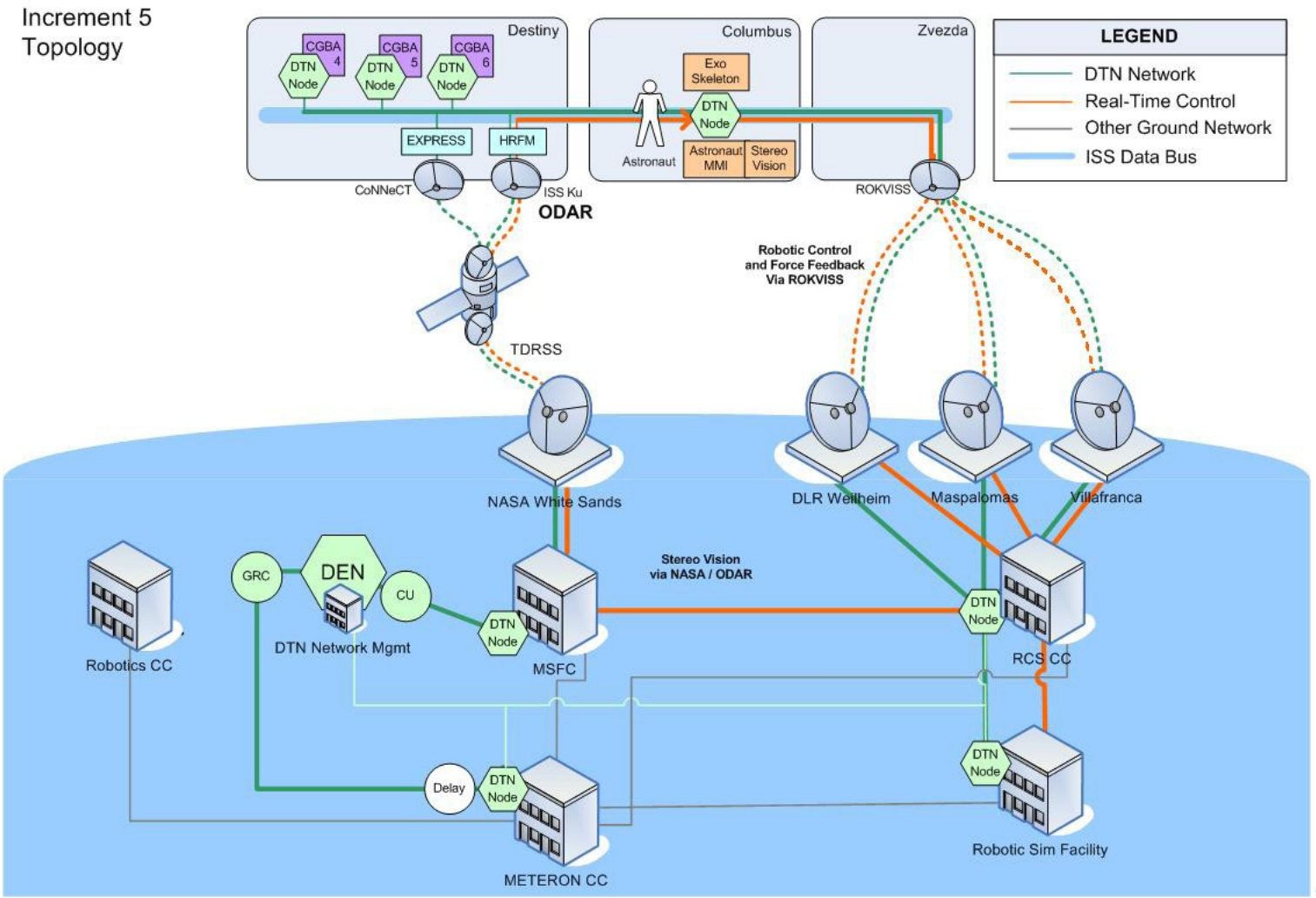
Interoperability

- Tested with the various DTN implementation in the middle of the Postellation DTN Cloud:
  - DTN2
  - IBR
  - ION
- Interop test plan from RFC5050 was created and applied against the implementations.

Exemple Scenario

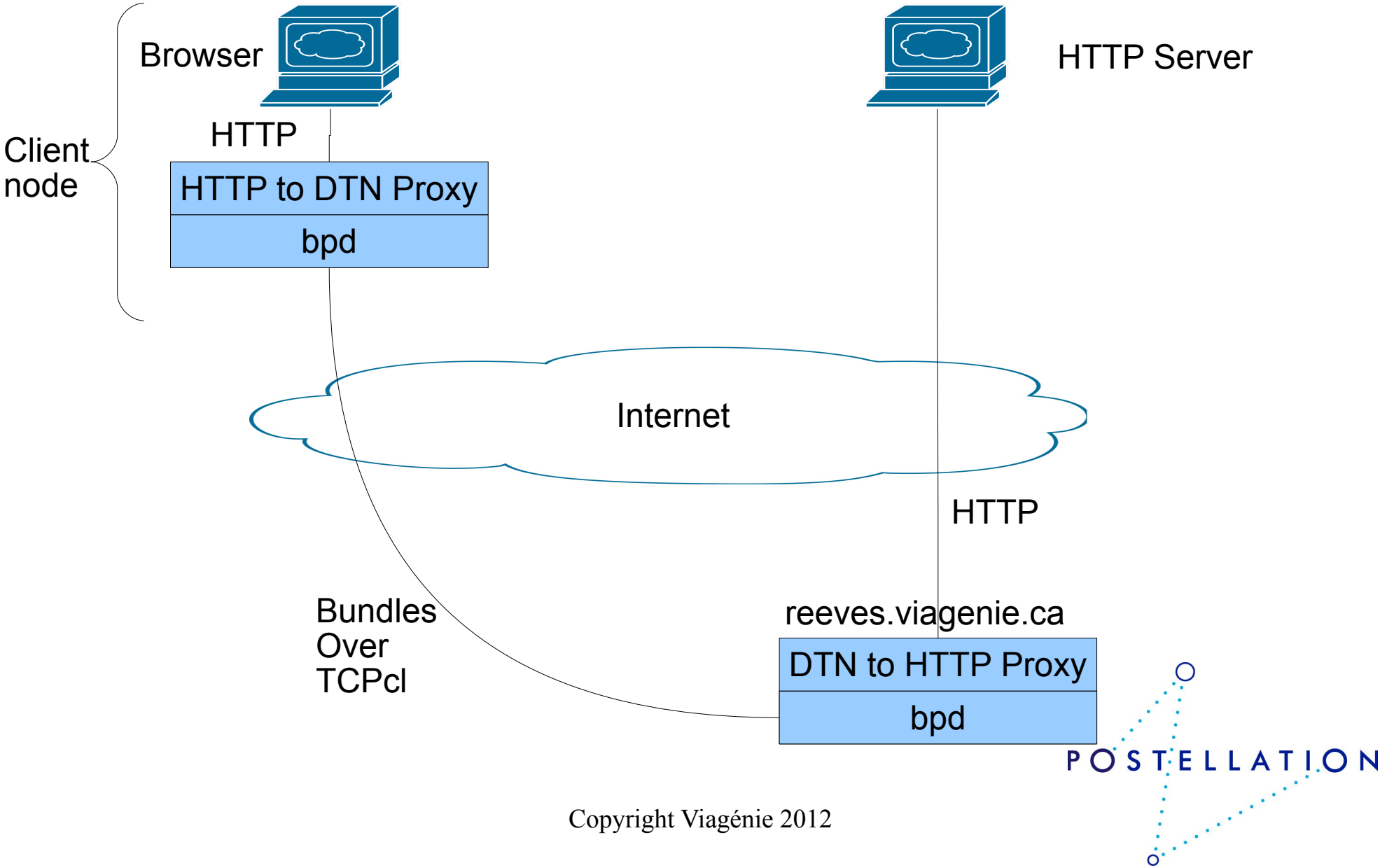


Meteron

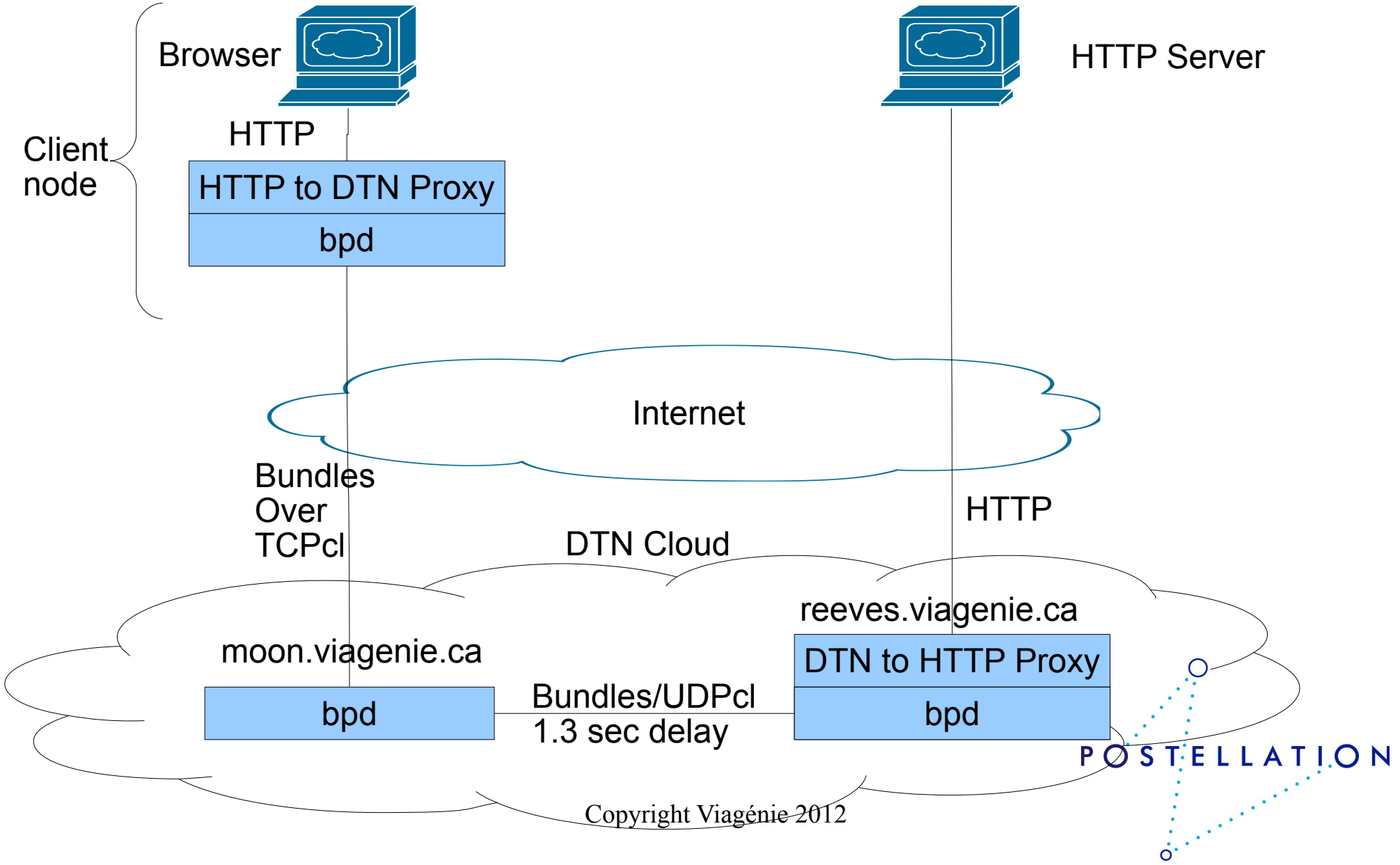


From: Multipurpose End-To-End Robotic Operations Network (METERON), ESA/NASA

Demonstration (1)



Demonstration (2)



Video Streaming

- If one uses Video streaming over http and carry it over a DTN, then the whole video will be buffered in the entry of the DTN network. Therefore, a latency as large as the length of the video will be seen by the end-user.
- Postellation optimize this by sending chuncks of video stream real-time, therefore the end-user will see almost no difference than full real-time. The only latency would be the actual latency of the DTN network itself.

Available to Try and Use

- Implementation:
  - has been tested in production work
  - connected automatically to the DTN node and HTTP proxy
- If you would like to test it out, go to:
  - http://postellation.viagenie.ca (via IPv4, IPv6 or DTN)
  - After downloading, uncompress, then run the “start” program. This will start Bundle Protocol, HTTP proxy and registers the node to the DTN network.
  - After running it, you can also subscribe to our RSS News Service Delivery over DTN, to receive your NASA news over DTN!

Porting to Real-Time Operating System

- System requirements, memory footprint for an i386 target running bpd:
  - Binary image size: 508 kB (full RTEMS OS + Postellation software)
  - Heap size: 256 kB (bare minimum for enabling the RTEMS networking stack)
  - Stack size: 4 kB (bare minimum on the i386 architecture)
- This shows that Postellation makes it possible to deploy a full DTN stack in under one megabyte of memory.

Conclusion

- **Lean** BP implementation → good for embedded systems
- Ported to most OS
- **Smart** http/https proxy for easy application deployment
- **Easy** deployment by automating registration and configuration
- Available to use: http://postellation.viagenie.ca

Questions?

Marc.Blanchet@viagenie.ca

This presentation: http://www.viagenie.ca/publications/

References

- http://postellation.viagenie.ca