A Delay-Tolerant Network Architecture for Challenged Internets

Kevin Fall

Presented by Ross Lagerwall

University of Cambridge

March 5, 2013

Background

TCP/IP expects performance characteristics from underlying links.

- An end-to-end path exists.
- Latency between any two nodes is not excessive.
- The probability of a packet being dropped is small.

Networks which violate these principles are called challenged networks.

Background

Challenged networks have the following characteristics:

- High latency and low bandwidth.
- Possibility of disconnection from a motion (such as a satellite movement) or a low-duty-cycle operation (such as with sensor networks).
- Long queuing times.
- Limited longevity.
- Low duty cycle.
- Limited resources.
- Interoperability challenges.
- Security challenges.

Solutions

Performance enhancing proxies and protocol boosters can be used to fool TCP/IP into working better over poor links.

Application-layer proxies provide a Internet-to-specialnet mapping.

But it is difficult to act as a traffic carrier and a traffic sink.

Email's asynchronous message delivery is a useful model but falls short in a few areas.

Architecture

A delay tolerant networking architecture (DTN) provides interoperability between challenged networks.

It operates as an overlay network so on the Internet, it will operate above TCP/IP whereas in space, it may use CFDP.

DTN consists of regions and DTN gateways which provide store-and-forward functionality.

They forward aggregate messages called bundles.

Unlike with ARPANET gateways, they perform reliable message routing rather than best effort packet switching.

Architecture

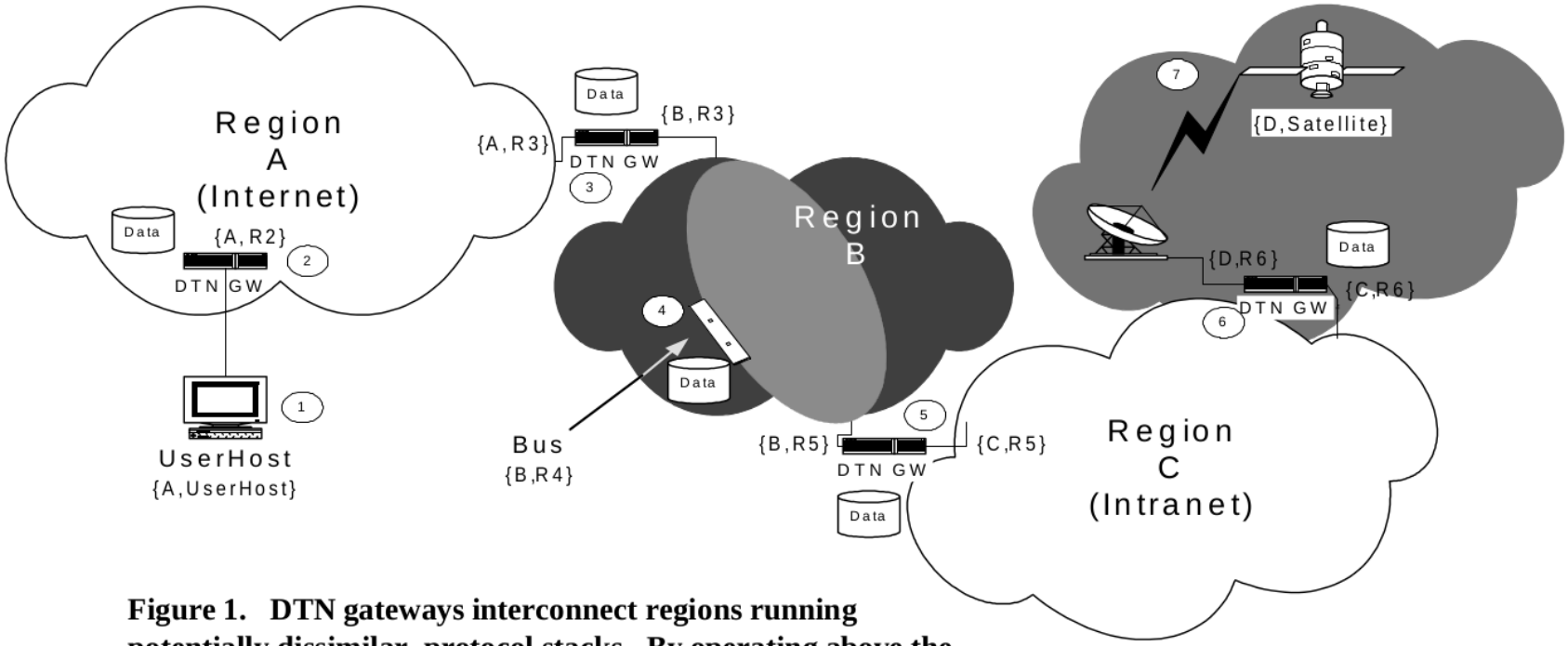

Figure 1. DTN gateways interconnect regions running potentially dissimilar protocol stacks. By operating above the transport protocols in use on the incident networks, they provide virtual message switching, in-network retransmission, and name mapping, allowing globally-interoperable names to be mapped

Architecture – Naming

Names consist of tuples of variable length.

`{internet.icann.int, "http://www.ietf.org/oview.html"}`

The first half of names are hierarchically structured and names the region.

The second half names the content within the destination region.

Naming uses late binding.

Architecture – Path Selection and Scheduling

Routes are made of contacts which are communication opportunities.

A contact has a number of parameters:

- Start and end times
- Capacity
- Latency
- Endpoints
- Direction

Problems include determining existence of contacts, knowing the state of pending messages and efficiently assigning messages to contacts.

Solving path selection and scheduling is for future work.

Architecture – Custody Transfer

Nodes can be persistent or non-persistent.

A non-persistent node can transfer custody of a message to a persistent node which then accepts responsibility for reliably delivering the message.

This is useful for the limited longevity situation and it means that nodes do not have to keep copies of their messages.

Delivery confirmation may be requested but it is application specific.

Architecture – Convergence Layers

The bundle forwarding function requires an underlying reliable delivery mechanism with message boundaries.

An implementation of DTN would require transport-layer specific convergence layers.

For example, TCP would require a convergence layer to add message boundaries.

Reliable delivery can, at worst, be implemented with timeouts and retransmission.

Architecture – Time Synchronization

DTN requires a form of time synchronization.

It is used for:

- Identifying message fragments.
- Purging messages which have exceeded their lifetimes.

The paper recommends time synchronization to on the order of 1ms.

It is questionable whether this would be possible in the varied networks that DTN is supposed to work with.

Architecture – Security

Each message contains:

- A verifiable identity of the sender.
- An approval of the requested class of service.
- A verification of the content of the message.

Routers and users obtain public and private keys.

A principal sends its signed public key and a signed message to the first gateway.

The first hop gateway checks the access control list and resigns the message.

Downstream gateways can then rely on this signature without checking needing to check the access control list.

Architecture – Congestion & Flow Control

Congestion control uses a priority queue for custody storage.

The queue is ordered by message priority and lifetime.

Messages which are too large to fit are rejected.

Priority inversion is a problem.

A DTN gateway uses the underlying network's flow control mechanism.

Application Interface

The DTN architecture provides a non-blocking API.

Applications must be designed not to expect timely responses.

Applications must be able to create and manipulate name tuples, class of service specifiers, and authentication information.

Opinion

The paper seems rather incomplete – the section on "Path Selection and Scheduling" is future work and the description of the API is inadequate.

There is a brief mention of an implementation in the conclusion but no evaluation of that implementation.

Each message contains two cryptographically verifiable pieces of information.

But, a DTN is supposed to work with nodes with extremely limited amounts of processing power and memory (such as sensor nodes).

A gateway needs to be able to do many cryptographic operations with a high message rate.