Topics:
- Internet Indirection Infrastructure
- DTNs (Delay-Tolerant Networks)
- The NewArch Project

Motivation:
- Despite intensive research, Internet lacks support for multicast, anycast, end-host mobility etc.
- Reason: basic assumption in Internet is one sender and one receiver placed at well-known and fixed location (point-to-point communication)
- Solution: indirection. Assumes an indirection point interposed between sender and receiver

Provide a rendezvous based communication abstraction (instead of point-to-point)
- Each packet is associated an identifier \( id \)
- To receive a packet with identifier \( id \), receiver \( R \) maintains a trigger \( (id, R) \) into the network

I3 is an overlay with a set of servers that store triggers and forward packets between end-points

To find a trigger that matches a given packet, I3 relies on a lookup service (Chord, Pastry, i.e. a P2P solution)

Whole procedure works as follows:
1. A trigger \( (id, addr) \) is stored at the server responsible for \( id \)
2. A packet \( (id, data) \) is forwarded based on \( id \) through the overlay network to that server
3. Then, the packet is matched to the trigger and forwarded to \( addr \) via IP
I3: public and private triggers

Identifiers of public triggers are known by all end-hosts in the system
- Public triggers are long lived (days or months)
- Example: trigger of a web server

Private triggers are chosen cooperatively by a small number of end hosts
- Short lived, exist only during the duration of a flow
- Example: client accessing a web server
  1. Client chooses a private trigger id(c) and sends it to the web server via the web server’s public trigger
  2. Server creates a private trigger id(s) and sends it to the client
  3. Client and server insert private triggers (id(c), addr(c)), (id(s), addr(s)) into I3, and use them to communicate
  4. When the communication terminates, the private triggers are destroyed

I3: Mobility

Host just needs to update its trigger as it moves from one subnet to another

I3: Multicast

Unifies multicast and unicast abstractions
- Multicast: receivers insert triggers with the same identifier

Figure from I. Stoica
Delay Tolerant Networks

Motivation:

TCP/IP suite can provide services when

- an end-to-end path between source and destination exits for the duration of the communication session
- (for reliable communication) the maximum RTT over that path is not excessive and not highly variable
- the end-to-end loss is relatively small
- Extreme environments → challenged networks → need for DTN

DTN: Extreme Environments 1

Some characteristics that make environments "extreme":

- Long and/or variable delays: affects interactivity, applications
  - Propagation delays (e.g. RTT Earth/Mars ~8 to ~40 minutes)
  - Queuing delays (e.g. slow processor due to power limitations)
  - Clocking delays (when a packet is transmitted only after full reception prior to transmission. Can give high per-packet delay in slow, multihop networks)

- Frequent partitioning (Intermittent reachability): adds to data loss
  - Can add to overall delay if nodes are configured to store data until the outbound link is restored

DTN: Extreme Environments 2

Data rate asymmetry: affects adversely protocols using ARQ for reliable delivery by altering the ACK return path

- Deep space communications: downlink data rate is intentionally engineered to be much greater than uplink data rate.

Packet loss and errors

- For very lossy environments, hop-by-hop retransmission is preferable to end-to-end retransmission

Examples of extreme environments

<table>
<thead>
<tr>
<th></th>
<th>Delay</th>
<th>Disconnection</th>
<th>Data rates</th>
<th>Data loss and error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired Internet</td>
<td>Short RTT</td>
<td>Rare</td>
<td>Very high</td>
<td>Low/none</td>
</tr>
<tr>
<td>IPN</td>
<td>10+ minutes</td>
<td>Weeks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTN</td>
<td></td>
<td>Frequent</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>SN (ocean env.)</td>
<td>Long</td>
<td>Long</td>
<td></td>
<td>Significant</td>
</tr>
</tbody>
</table>
DTN: Principles of design 1

Use transport and network layer technologies appropriate to the local environment
- DTN should provide the means for dissimilar networks to interoperate
  - Single end-to-end transport protocol out of the question
- Whereas IP provides the common substrate for the Internet above dissimilar link layers, the DTN architecture creates a network of Internets by providing an end-to-end layer above the transport layer, called the bundle layer.

Employ a “non-chatty” communications model
- Avoid interactive styles of communication
- Entire application-layer messages, called bundles, move through the network as units.

DTN: Principles of design 2

Base transfers between nodes on store-and-forward techniques
- IP based on store-and-forward transmission
  - “store” last only long enough to determine next hop and wait for packets ahead in the queue to be transmitted. The packet is typically discarded if no route to destination exists at the time the routing engine examines the packet.
- DTN: no path to destination at time of bundle receipt is normal
  - Fundamental assumption for DTN: considerable amount of storage is available in the network.

Advance the point of retransmission toward the destination
- IP: retransmission of loss data is end-to-end
- DTN: advance the point of data retransmission toward the destination whenever possible
  - Earlier release of retransmission resources
  - Retransmitted data arrives sooner

DTN: Architectural overview 1

DTN architecture is based on store-and-forward message switching
- DTN transmits bundles that contain both user data and relevant metadata. Bundles = “messages” on which DTN architecture is based
- The bundle layer stores and forwards bundles between nodes
- A single bundle layer is used across all networks (regions) that make up a DTN
- The layers below the bundle layer (transport and below) are chosen for their appropriateness of each region (Internet vs. sensor networks)
- A message-switched architecture gives the network a priori knowledge of the size and performance requirements of requested data transfers.
  - Significant advantage when queuing is considerable

DTN: Architectural overview

- Application
- Bundle
- Transport (TCP)
- Network (IP)
- Link
- Physical

Internet Layers

DTN Layers

common across all DTN regions

specific to each DTN region
DTN Nodes

In DTN, a node is an entity with a bundle layer. A node can be:

- A host: sends and/or receives bundles, but does not forward them. The bundle layer of a host that operates over long-delay links requires persistent storage.
- A router: forwards bundles within a region. The bundle layer of a router that operates over long-delay links requires persistent storage.
- A gateway: forwards bundles between two or more DTN regions. Gateways provide conversions between the lower-layer protocols of the regions they span.

DTN: Custody transfers

End-to-end reliability can only be implemented at bundle layer

- No single transport-layer operates end-to-end.

When the current bundle-layer custodian sends a bundle to the next node, it requests a custody transfer and starts a timer.

- A bundle custodian must store a bundle until either (1) another node accepts the custody, or (2) expiration of the bundle’s time-to-live.

Note: custody transfers do not provide guaranteed end-to-end delivery

- “return receipt”: confirmation to source that bundle has been received by destination application.

Custody transfers together with a reliable transport-layer move the point of retransmission towards the destination.

- For lossy paths, this can be beneficial.
DTN: Regions

- **DTN** = "network of Internets"
- Each DTN region has a unique identifier that is known (or knowable) among all regions of the DTN
- All inter-region communication takes place via DTN gateways
- DTN gateways vs ARPANET gateways
  - Both provide conversions between protocols specific to diff regions
  - But DTN gw operate above transport layer and switch messages

DTN: Status and Perspectives

**Status:**
- DTN architecture is being developed
- In March 2003, source code was released
- Deployment scheme with five regions has been "sketched"

**Perspectives:**
- DTN incorporates delay and disconnectedness
- Good for, e.g., sensor networks, deep ocean communication etc.
- But think also of users traveling with their laptops.

The NewArch project

NewArch: A new architecture for an Internet

Among the participants:
David D. Clark (MIT), Bob Braden (ISI), Mark Handley (ACIRI), Scott Shenker (ACIRI), John Wroclawski (MIT)

Lots of papers on future direction for the Internet, see [http://www.isi.edu/newarch/](http://www.isi.edu/newarch/)

Literature

Internet Indirection Infrastructure
Also: First workshop on hot topics in Networks (Hotnets-I)

Delay Tolerant Networking Research Group (material here taken mostly from the tutorial and from Juan Alonso, thanks!):

The NewArch project (lots of papers):
[http://www.isi.edu/newarch/](http://www.isi.edu/newarch/)