Pervasive Systems

Ioannis Chatzigiannakis
Sapienza University of Rome
Department of Computer, Control, and Management Engineering (DIAG)

Lecture 4:
Delay-Tolerant Networking

IP-based Network Assumptions

- End-to-end RTT is not terribly large.
  - A few seconds at the very most – typically less than 500ms,
  - window-based flow/congestion control works.
- Some path exists between endpoints.
  - Routing finds single “best” existing route.
- E2E Reliability using ARQ works well.
  - True for low loss rates (under 2% or so).
- Packet switching is the right abstraction.
  - Internet/IP makes packet switching interoperable.

Typical IP-based Network E2E Path

Non-IP-based Networks

- Stochastic mobility
  - Military/tactical networks
  - Mobile routers w/disconnection (e.g. ZebraNet)
- Periodic/predictable mobility
  - Spacecraft communications
  - Busses, mail trucks, police cars, etc. (Info Stations)
- “Exotic” links
  - Deep space [40+ min RTT; episodic connectivity
  - Underwater [acoustics: low capacity, high error rates & latencies]
Delay Tolerant Networking
Routing Protocols
Deployments
Introduction
Delay-Tolerant Network E2E Path

Ioannis Chatzigiannakis
Pervasive Systems
Lecture 4 5 / 41

New Challenges

- Very Large Delays
  - Natural delay could be seconds to minutes
  - If disconnected, may be (effectively) much longer
- Intermittent/Scheduled/Opportunistic Links
  - Scheduled transfers can save power and help congestion; scheduling common for esoteric links
- High Link Error Rates / Low Capacity
  - RF noise, light or acoustic interference
- Different Network Architectures
  - Many specialized networks won’t/can’t ever run IP

Ioannis Chatzigiannakis
Pervasive Systems
Lecture 4 6 / 41

How to Address these issues?

- Some problems surmountable using Internet/IP
  - “cover up” the link problems using PEPs
  - Mostly used at “edges”, not so much for transit
- Performance Enhancing Proxies (PEPs):
  - Do “something” in the data stream causing endpoint (TCP/IP) systems to not notice there are problems
  - Lots of issues with transparency security, operation with asymmetric routing, etc.
- Some environments never have an E2E path
  - Consider an approach tolerating disconnection, etc...

Ioannis Chatzigiannakis
Pervasive Systems
Lecture 4 7 / 41

Delay and Disruption Tolerant Networking (DTN)

- Support interoperability across radically heterogeneous networks
- Acceptable performance in high loss/delay/error/disconnected environments
- Decent performance for low loss/delay/errors
- Environments without continuous network connectivity.
- For challenged environments: remote sensors in Antarctica, a spacecraft in deep space, submersible vessels etc.
- For communication during Disasters and Emergency.
- Communication is asynchronous by nature.
Delay-Tolerant Network Store & Forward Approach

To enable communication between source and destination without the support of a fixed network infrastructure.
# Message Overlay Abstraction

- Message exchanges are called **bundles**.
  - "postal-like" message delivery over regional transports with coarse-grained CoS [4 classes]
  - Options: return receipt, "traceroute"-like function, alternative reply-to field, custody transfer
  - Supportable on nearly any type of network
- Bundles are routed in a store and forward manner.
  - "Application data units" of possibly-large size
  - May require framing above some transport protocols
  - API supports response processing long after request was sent (application re-animation)

---

## Is it an e-mail like communication scheme?

- Many similarities to (abstract) e-mail service
- Primary difference involves routing and API
- E-mail depends on an underlying layers routing:
  - Cannot generally move messages closer to their destinations in a partitioned network
  - In the Internet (SMTP) case, not disconnection-tolerant or efficient for long RTTs due to "chattiness"
- E-mail security authenticates only user-to-user

---

## Store, Carry and Forward Scheme for Routing

- A node stores a message until an appropriate communication opportunity arises.
- A series of independent forwarding decisions.
- Eventually bring the packet to its destination.
- Key decisions in forwarding packets:
  - What to send (own packet or a relayed packet ?)
  - To whom (to a relay or the destination ?)
  - When to do so (will suffer collisions, or cause interference ?)
- Simple (and efficient) approach: Randomize on
  - Whom to send,
  - When to send.
### Store, Carry and Forward

**Strategies**

- **Single-Copy**: only a single copy of each message exists in the network at any time.
  - lower number of transmission,
  - lower contention for shared resources.

- **Multiple-Copy**: multiple copies of a message may exist concurrently in the network.
  - lower delivery delay,
  - higher robustness.
**Single-Copy Strategy**

- Direct transmission – Source only forwards message to destination.
- First Contact – Node A forwards message to the first encounters.
- Randomized routing – Node A forwards message to node B with probability \( p \).

**Multi-Copy Strategy**

- Epidemic routing
  - Spread of “disease”.
  - Whenever “infected” node meets “susceptible” node, a copy is forwarded.
- Utility-based routing:
  - Node A forwards a message to node D to node B iff \( U_A(D) < U_B(D) \)

**Basic Idea of Epidemic Routing**

- Bio-inspired: packets are considered to infect nodes (Vahdat & Becker, 2000)
- Nodes are randomly mobile & have ordered identifiers.
- Resources sufficiency (battery / buffers).
- Forwarding Decision: fixed – flooding
- Buffers: FIFO
- Buffer (hashed) “index”: Summary Vector (SV)
- Reliability: acks

**Main Mechanism**

- Upon meeting a newly identified neighbor node
  - Exchange SVs,
  - Exchange unknown messages.
- For protocol sake the process is initiated by the node with the smaller identifier.
- Per-host queuing.
- New messages given preference over old ones in terms of buffer availability.
Main Concepts of PRoPHET Protocol

- Users move in a “not so random”, predictable fashion.
- Forwarding decision: by Delivery Predictability $P(M, D)$ set up at every node $M$ for each known destination $D$.
- Epidemic Routing SVs are used here too to exchange.

Protocol Details

- When the node $M$ encounters another node $N$, the predictability for $N$ increases a
  \[ P(M, N)_{new} = P(M, N)_{old} + (1 - P(M, N)_{old}) \times L_{enc} \]
  where $L_{enc}$ is an initialization constant.
- The predictabilities for all destinations $D$ other than $N$ suffer ageing:
  \[ P(M, D)_{new} = P(M, D)_{old} \times \gamma \times K \]
  where $\gamma$ is an aging constant, and $K$ is a time factor.

Main concepts of MAXPROP

- Motivated by pedestrian mobility and city vehicles (busses) (Burgess, et al. 2006)
- Addressed resources issues considering vehicles.
  - Bulky equipment,
  - Energy.
- Maintains ordered destination based queues.
  - Addresses on top of PRoPHET.
  - QoS,
  - Stale data.
- Assumes:
  - Unlimited buffer for own messages per node,
  - Fixed size buffer for relaying messages,
  - No topology knowledge/control.
Communication steps of MAXPROP

- Neighbor Discovery.
  - no knowledge of when the next opportunity to communicate will be.
- Data Transfer.
  1. Transfer packets destined for neighbor peer,
  2. Transfer routing information,
  3. Acknowledge any delivered data,
  4. Prioritize "young" relayed packets,
  5. Send un-transmitted packets by estimated delivery likelihood,
  6. Ensure only new packets are sent.
- Storage Management.
  - Expunge packets to accommodate the relay buffers.

Which Routing Schemes is more suitable?

- Design space is LARGE
- Many different protocols have been proposed in the literature

Diverse Features

Node B

An DTN Protocol Architecture

A DTN protocol scheme is made of the combination of three basic techniques:

- Queue management (QM)
- Replication(R)
- Forwarding(FW)
How to Evaluate Routing Schemes?

- Many factors affect performance
  - Mobility of nodes,
  - Size of area covered by nodes,
  - Available Resources,
  - Network Traffic.

Quantitative Evaluation using Simulation

Software Simulation Parameters

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Time</td>
<td>12 hours</td>
</tr>
<tr>
<td>World Size</td>
<td>400x200 m²</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>10, 20, 40, 70</td>
</tr>
<tr>
<td>Message Size</td>
<td>100B-200B</td>
</tr>
<tr>
<td>Message Creation Interval</td>
<td>60 - 180s</td>
</tr>
<tr>
<td>Transmission speed</td>
<td>25 kbps</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m</td>
</tr>
<tr>
<td>Node speed</td>
<td>0.8-2.0 m/s</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>10KB, 100KB</td>
</tr>
</tbody>
</table>

3-4 Protocols from each Class

Evaluation Result: Delivery Ratio
Evaluation Result: Overhead

Wildlife Monitoring

- The users are the biologists.
- Track position of zebras in wildlife.
- Special collars with GPS are attached to zebras.
- Tracking data is replicated when animals are in reach of each other.
- Tracking data gathered daily or weekly using a base station in a car or plane (called a "message ferry").
- Project did not use the term “DTN”.

Prototype Hardware

Prototype Hardware

![Prototype Hardware Images]

MaxStream Wireless Data Modem

U-blox GPS-MS1E (GPS/CPU)
Technical Considerations

- Track animals long term and over long distances.
- All sensing nodes are mobile.
- Large area: 100s ... 1000s sq kilometers.
- “Coarse-Grained” nodes.
- GPS on-board.
- Long-running and autonomous.

Public Transportation Network

- A DTN over public transportation busses.
- Deployed in Amherst and surrounding county.
- Includes 40 busses.
- Network inaccessibility corresponds to physical inaccessibility.
  - DTN administration difficult,
  - DTN system administration must be accomplished in a disruption-tolerant manner.
- DTN solution handles configuration of IP, Link, and physical network.
- Buses transfer data as they pass by each other and via available hot spots.

The OpenBrick Hardware
The OpenBrick Hardware
Software Considerations

- Autonomous operation and “shutdown”.
- Maximal failure tolerance.
  - No remote administrator login or local terminal.
  - Physical inaccessibility during operation (6am - 1am).
  - Nodes versus support staff.
- Handle network interface configurations.
  - Support IP programs.
- Handle bundle routing.
  - Support DTN programs.

Software Components

- Init and Restore
  - Perl and Python
- Auto update
  - Python
- Live IP
  - Java 1.4
- GPS update
  - Python
- DTN daemon
  - Java 1.4 and linux-only Java 1.3

Connection Events

- Red dots indicate bus-to-bus transfers (1 month)
- Measurement of real TCP throughput.
- Traces collected everyday.