Routing in a Delay Tolerant Network

Vladislav Marinov

Jacobs University Bremen

March 31st, 2008
Dial-up connection - 32 kbps, available between 11pm and 6am (otherwise very expensive)

Digital courier with a USB storage device - 1 Mbps, available 3 times a day for a period of 5 minutes, storage limit of 128MB, trip to the city takes 2 hours

Satellite - 80 kbps, visible 4 times a day for a period of 10 minutes, storage limit of 500MB
Outline

1. The Delay Tolerant Network Model
2. Routing in a DTN
3. DTN Routing Algorithms Framework
4. Performance Evaluation
Outline

1. The Delay Tolerant Network Model
2. Routing in a DTN
3. DTN Routing Algorithms Framework
4. Performance Evaluation
Existing TCP/IP Internet Model

- Packet-switched model of service
- Interoperability due to a layered approach (physical, data-link, network etc.)
- The IP layer can be mapped on a number of dissimilar data link layer frames

Assumptions:

- An end-to-end path between a source and a destination always exists
- The maximum round trip time between any two nodes is not excessive
- Small end-to-end packet drop probability
Delay Tolerant Network (DTN) Model

- An overlay architecture at the application layer using message-based switching[1]

- Networks, which may experience frequent, long-duration partitioning and may never have an end-to-end contemporaneous path.

- Link capacities and propagation delays are time-dependent - a DTN can be modelled as a directed multi-graph in which edges are characterized by $e_n = ((u, v)_n, c(t), d(t))$

- The opportunity to send data over an edge is called contact.

- Communication is performed via messages represented by a tuple ($src$, $dst$, $time$, $size$).

- Nodes have finite-long term storage for holding application data and in-transit data.
1. The Delay Tolerant Network Model
2. Routing in a DTN
3. DTN Routing Algorithms Framework
4. Performance Evaluation
Routing occurs in a store and forward fashion - messages may be stored for long time at intermediate nodes until a contact becomes available

- In traditional TCP/IP Internet if the next hop is not available the packet is discarded

The routing algorithm is responsible for determining the next edge that a message should be forwarded along

The DTN routing problem can be viewed as a constrained optimization problem

- Minimize delay of a message (routing metric)
- Some edges might be unavailable for an extended period of time
- Each node has limited storage resources
1. The Delay Tolerant Network Model
2. Routing in a DTN
3. DTN Routing Algorithms Framework
4. Performance Evaluation
Knowledge Oracles

- Contacts Summary Oracle - can answer questions about aggregate statistics of contacts i.e what is the average waiting time until the next contact of an edge
- Contacts Oracle - can answer any question about contacts between any two nodes at any point of time
- Queuing Oracle - gives information about buffer occupancies at any node at any point of time
- Traffic Demand Oracle - can answer any question about messages injected into the DTN at any point of time
Routing with Zero Knowledge

- Algorithm: **First Contact (FC)**
- Does not utilize any oracles
- Idea: Choose the next hop at random
- Performs poorly in non-trivial topologies because forwarding along a selected edge may not make any progress towards a destination, lead to a dead end or form a loop
Algorithm with Time-Invariant Costs

- Algorithm: **Minimum Expected Delay (MED)**
- Oracles: Contacts Summary
- Proactive routing approach i.e static routes
- Use Dijkstra to compute the routes and costs
- The edge cost is the sum of the average waiting time, the average propagation delay and the average transmission delay
- Minimizes average waiting time but fails to exploit superior edges that become available after the route has been computed
  - Imagine a direct contact to the message destination that becomes available while the message is waiting for its statically precomputed next-hop
Use Dijkstra with time-varying costs

- \( m \) - message size
- \( c(e, t) \) - capacity of the DTN
- \( d(e, t) \) - propagation delay of the DTN
- \( Q(e, t, s) \) - queue size at the source of edge \( e \), at time \( t \) as predicted by node \( s \)

\( \omega(e, t) = \omega'(e, t, m, s) \)

\( t'(e, t, m, s) = \min\{ t'' | \int_{x=t}^{t''} c(e, x) \geq (m + Q(e, t, s)) \} \) - the earliest time the message at edge \( e \) and the queued data can be unloaded into the network

\( \omega'(e, t, m, s) = t'(e, t, m, s) - t + d(e, t') \) - captures the total transfer time of a message with size \( m \) over the edge \( e \)
Algorithms with Time-Varying Costs

- **Earliest Delivery (ED)**
  - Oracles: Contacts
  - \( Q(e, t, s) = 0 \) - no queuing incorporated
  - Source routing approach
  - No consideration for storage and queuing at intermediate nodes - messages might miss a contact or get dropped because of buffer overflow
  - Optimal when there are no queued messages or when contact capacity is large
Algorithms with Time-Varying Costs

- **Earliest Delivery with Local Queueing (EDLQ)**
  - Oracles: Contacts
  - 
    \[
    Q(e, t, s) = \begin{cases} 
    \text{data queued for } e \text{ at time } t & \text{if } e = (s, \ast) \\
    0 & \text{otherwise}
    \end{cases}
    \]
    (1)
  - Per-hop routing approach
  - Avoids congestion at the source, but does not account for queueing at intermediate nodes
  - Messages might get dropped because of buffer overflow

- **Earliest Delivery with All Queues (EDAQ)**
  - Oracles: Contacts and Queues
  - \( Q(e, t, s) = \) data queued for \( e \) at time \( t \) at node \( s \)
  - Source routing approach
  - Takes care of congestion in the whole DTN by making bandwidth reservations
Outline

1. The Delay Tolerant Network Model
2. Routing in a DTN
3. DTN Routing Algorithms Framework
4. Performance Evaluation
San Francisco Bus Network

- 20 buses capable of DTN store and forward operations
- The buses are able to exchange messages when in a radio range (default is 100m)
- The simulations continues for 24 hours and is split in intervals of 1 hour
- During each interval 20 random bus source/destination pairs are chosen
- Each bus sends 200 messages to its destination in each interval
- Default storage capacity of each bus is 100MB
- Default link bandwidth is 100KBps
- Evaluate the performance of the routing algorithms by varying chosen parameters
- Performance metrics: average delay and delivery ratio
Varying Bandwidth

- Load = (traffic demand) / (total amount of traffic that can be carried by the DTN)
- Increasing the bandwidth increases the total amount of traffic that can be carried and thus decreases the load
- At small load time-varying algorithms perform better (ED, EDLQ, EDAQ)
- As load increases ED starts to perform like MED because of missed contacts (ED does not take into account queuing)
- At high loads the network becomes saturated so all algorithms perform similarly
Varying Bandwidth

Average Delay (hours)

Load

FC
MED
ED
EDLQ
EDAQ
Varying Range

- Increasing the radio range increases the total amount of traffic that can be carried by a DTN since nodes are in contact more frequently.
- Increasing the range reduces the average delay.
- Time-varying algorithms (ED, EDLQ, EDAQ) perform are much smarter at small ranges.
- As the range gets bigger and bigger the benefits of ED, EDLQ, EDAQ vanish because the nodes are almost always in contact and thus even the random FC is capable of routing messages successfully.
Varying Range

Average Delay (hours)

Radio range (meters)

FC
MED
ED
EDLQ
EDAQ
Conclusion

- Routing in a DTN is performed using the store-and-forward method and can be viewed as a constrained optimization problem.
- A number of routing algorithms were presented and their performance was compared in a simulation.
- In networks with plentiful communication opportunities the need for smart routing algorithms is minimal.
- In networks where resources are limited (contact opportunities, bandwidth or storage) smarter algorithms (ED, EDAQ, EDLQ) provide significant benefit.
- Algorithms which take network congestion into account (EDLQ, EDAQ) perform much better in extreme environments.
Kevin Fall.
A delay-tolerant network architecture for challenged internets.

Sushant Jain, Kevin Fall, and Rabin Patra.
Routing in a delay tolerant network.
Utilize modified Dijkstra

Input: $G = (V, E)$, $s$, $T$, $w(e,t)$
Output: $L$

1: $Q \leftarrow \{V\}$
2: $L[s] \leftarrow 0$, $L[v] \leftarrow \infty \forall v \in V \ s.t \ v \neq s$.
3: while $Q \neq \{\}$ do
4: Let $u \in Q$ be the node s.t $L[u] = \min_{x \in Q} L[x]$
5: $Q = Q \leftarrow \{u\}$
6: for each edge $e \in E$, s.t. $e = (u, v)$ do
7: if $L[v] > (L[u] + w(e, L[u] + T))$ then
8: $L[v] \leftarrow L[u] + w(e, L[u] + T)$
9: end if
10: end for
11: end while