ExOR: Opportunistic Multi-Hop Routing for Wireless Networks

Sanjit Biswas and Robert Morris
M.I.T. Computer Science and Artificial Intelligence Laboratory

Adapted from “CS7260 slides” by Nick Feamster, Georgia Tech
Presented by Mikhail Afanasyev
Traditional routing

- Identify a route, forward over links
- Abstract radio to look like a wired link

ExOR Slides adapted from http://pdos.csail.mit.edu/papers/roofnet:exor-sigcomm05/
Radios aren’t wires

- Every packet is broadcast
- Reception is probabilistic
Decide who forwards after reception

**Goal:** only closest receiver should forward

**Challenge:** agree efficiently and avoid duplicate transmissions
Why ExOR might increase throughput

- Best traditional route over 50% hops: $3^{(1/0.5)} = 6$ tx
- Throughput $\cong \frac{1}{\text{# transmissions}}$
- ExOR exploits lucky long receptions: 4 transmissions
- Assumes probability falls off gradually with distance
Why ExOR might increase throughput

- Traditional routing: $\frac{1}{0.25} + 1 = 5$ tx
- ExOR: $\frac{1}{(1 - (1 - 0.25)^4)} + 1 = 2.5$ transmissions
- Assumes independent losses
Batch Maps

- Challenge: finding the closest node to have rx’d
- Send batches of packets for efficiency
- Node closest to the dst sends first
  - Other nodes listen, send remaining packets in turn
- Repeat schedule until dst has whole batch
Reliable summaries

- Repeat summaries in every data packet
- Cumulative: what all previous nodes rx'd
- This is a gossip mechanism for summaries
Goal: nodes “closest” to the destination send first
Sort by ETX metric to dst
- Nodes periodically flood ETX “link state” measurements
- Path ETX is weighted shortest path (Dijkstra’s algorithm)
Source sorts, includes list in ExOR header
Details in the paper
ExOR Evaluation

- Does ExOR increase throughput?
- When/why does it work well?
25 Highest throughput pairs

<table>
<thead>
<tr>
<th>Node Pair</th>
<th>Throughput (Kbits/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExOR</td>
<td></td>
</tr>
<tr>
<td>Traditional</td>
<td></td>
</tr>
<tr>
<td>1 Traditional Hop</td>
<td>1.14x</td>
</tr>
<tr>
<td>2 Traditional Hops</td>
<td>1.7x</td>
</tr>
<tr>
<td>3 Traditional Hops</td>
<td>2.3x</td>
</tr>
</tbody>
</table>
25 Lowest throughput pairs

<table>
<thead>
<tr>
<th>Node Pair</th>
<th>4 Traditional Hops</th>
<th>3.3x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExOR</td>
<td>Traditional Routing</td>
<td></td>
</tr>
</tbody>
</table>

Throughput (Kbits/sec)

Longer Routes

Node Pair
ExOR uses links in parallel

Traditional Routing
3 forwarders
4 links

ExOR
7 forwarders
18 links
ExOR moves packets farther

- ExOR average: 422 meters/transmission
- Traditional Routing average: 205 meters/tx
Using ExOR with TCP

Batching requires more packets than typical TCP window
Questions?
Efficiency through Eavesdropping: Link-layer Packet Caching in Mesh Networks

Mikhail Afanasyev, David G. Andersen†, and Alex C. Snoeren
University of California, San Diego and †Carnegie Mellon University
Taking Advantage of Overhearing

• Wireless overhearing has a great potential
• Hard to take advantage of:
  – Complicated protocols
  – Topology requirements
  – Traffic requirements
  – Large latencies
  – Incompatibility with standards
Solution

Ask receiver before each packet

- Hop-over transmissions
- Retransmissions
- Repeated data
RTS-id
Re-use 802.11 RTS/CTS mechanism

Sender

RTS +data id
CTS
DATA id=8
ACK
RTS +data id
CTS-ACK

Receiver

Packet cache
#8
Keeping it compatible

Request To Send:

Clear To Send:

0 = CTS-ACK
Adaptive enabling

Some overhead - not needed on all links
Keep cache as usual, send results in ACKs
Simulation

Use Roofnet dataset
Outdoor mesh network with many opportunities for overhearing
Simulation Results

![Graph showing simulation results with various transmission savings. The graph plots the number of paths against transmission savings. The legend includes categories such as Overall, 2 hops, 3 hops, 4 hops, and >4 hops.]
Simulation Results

![Graph showing simulation results with different path transmission metrics: Shortest Path, RTSID Shortest Path, ETX, RTSID ETX, ETT, and RTSID ETT. The x-axis represents the number of transmissions per path, and the y-axis represents the number of paths. The graph compares the performance of these metrics under varying conditions.]
Implementation

- Implemented on CalRadio platform
- Open-source 802.11b radio
- 50MHz CPU with Linux
- 100MHz DSP (Digital Signal Processor)
- Baseband chip from Prism
- Highly customizable MAC
- This project is first user of the radio
Implementation Results

<table>
<thead>
<tr>
<th>Node</th>
<th>Rx (Over)</th>
<th>CTS-ACK</th>
<th>CTS</th>
<th>T.O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>0%</td>
<td>66.4%</td>
<td>33.6%</td>
</tr>
<tr>
<td>B</td>
<td>66.4% (0)</td>
<td>91.0%</td>
<td>5.6%</td>
<td>3.4%</td>
</tr>
<tr>
<td>C</td>
<td>66.4% (96.1%)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Avg. data trans. per packet w/ RTS-id: 1.125

<table>
<thead>
<tr>
<th>Node</th>
<th>Rx (Over)</th>
<th>CTS-ACK</th>
<th>CTS</th>
<th>T.O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
<td>78.1%</td>
<td>21.9%</td>
</tr>
<tr>
<td>B</td>
<td>78.1% (-)</td>
<td>-</td>
<td>96%</td>
<td>4%</td>
</tr>
<tr>
<td>C</td>
<td>75% (-)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Avg. data trans. per packet w/o RTS-id: 2.0
Questions?