Project 5
Ad-Hoc Networking

Ayush Dubey

Slide heritage: Bernard Wong → Robert Escriva → Zhiyuan Teo

Cornell CS 4411, November 2, 2012
Announcements

- Project 4 due Saturday at 11:59 PM.
- Project 5 will be released after Saturday, due in a week from release date.
- Web page is being updated frequently; check for updates.
There are three kinds of death in this world. There’s heart death, there’s brain death, and there’s being off the network.

Guy Almes
1. The 1,000 Foot Picture
2. Project Scope
3. Implementation
4. Concluding thoughts
What is an “ad-hoc networking layer”?
What is an “ad-hoc networking layer”?

Ad-hoc networking enables wireless communication without the need for infrastructure.

What is it useful for?

- Removes infrastructure costs.
- Allows quick deployment.
- Potentially more reliable (no single point of failure).

Based on Dynamic Source Routing.*

*http://www.cs.cornell.edu/People/egs/615/johnson-dsr.pdf
What do you mean by routing?

- Packets that arrive at your machine may not be meant for you.
- Packets not meant for you should be routed to their destination.
- Add a routing layer between the network and transport layer.
- Both minimsg and minisocket implementations should work on top of miniroute.
Our networking stack till now

- User Application
- Minimsg Header
- Minisocket Header
- Network
Our networking stack after P5

User Application

<table>
<thead>
<tr>
<th>Minimsg Header</th>
<th>Minisocket Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miniroute</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
</tbody>
</table>
The new header in pictures

User Application
Minisocket Header
Minimsg Header
Miniroute Header
Network
How does DSR work?

- DSR is a reactive protocol.
- When a host does not know the route a packet, it must discover the route.
  - It does so by sending a route discover packet†.
- A route discover packet is broadcast to all hosts within proximity of a wireless signal.
- Hosts will re-broadcast discovery packets if they are not the destination.
  - The host will add itself to the constructed route.
- The destination will send a route reply packet along the reverse route.

† Also called route request packet
DSR in pictures
DSR in pictures: RREQ (A $\rightarrow$ G)
DSR in pictures: RREQ (A → G)
DSR in pictures: RREQ (A → G)
DSR in pictures

[A; B; E; G]
DSR in pictures: RREP
DSR in pictures: RREP
DSR in pictures: RREP
DSR in pictures: RREP

A — C — D
B — E — G
F — H

G; E; B; A
Data, Route = [A; B; E; G]
Routing protocol (common case)

- If a route contains $A \rightarrow B \rightarrow C$ where $A$ is the source, $C$ is the destination, $C$ will flip the route to $C \rightarrow B \rightarrow A$ to send a reply to host $A$.
- Upon receiving a reply, $A$ will add this route into its route cache, and use it to send data.
- Route cache entries expire in 3 seconds to prevent stale cache entries.
- Route discovery must be performed again upon route expiration.
  - Is there a better way to do this?
Routing protocol (failure conditions)

- How does the protocol terminate?
  - By tracking a time-to-live (\texttt{ttl}) value.
- What should the \texttt{ttl} be initialized to?
- What happens when \texttt{ttl} = 0?
- How do we prevent loops?
What do you need to implement?

- **Change calls to** `network_send_pkt` **to** `miniroute_send_pkt`.
  - But your `miniroute_send_pkt` function may still use `network_send_pkt` for unicasts.

- **Update the network handler.**
  - Interpret the miniroute header.
  - Handle routing control packets.
  - Deliver `minimsg/minisocket` packet as usual if the destination has been reached.
Routing Cache

- The cache must be able to hold \texttt{SIZE_OF_ROUTE_CACHE} entries.
- Old items are invalidated after timeout.
  - Alarms may be used, but it can be done without.
- Cache access should be efficient, as you may increase \texttt{SIZE_OF_ROUTE_CACHE} to be large.
  - Aim for $O(1)$ cache operations.
  - Our suggestion – hash table (we have provided a hash function for network addresses).
miniroute_send_pkt requirements

- Allow only one route discovery request per destination on the network at any one time.
  - Block threads if `miniroute_send_pkt()` was called and route isn’t in the cache.
  - Multiple threads should not trigger multiple routing discovery requests for the same destination.
  - Unblock all threads waiting on a route when that reply arrives.
Miniroute packets

- Use the header format provided in `miniroute.h`
  - Routing interoperability requires headers be in network byte order – use same packing/unpacking functions as before.
  - `MAX_NETWORK_PKT_SIZE` still the same – may have to change P3/P4 code.
- Try running `network6.c` over `miniroute`
  - Test interoperability with friends.
Additional implementation requirements

- Need to track recently seen discovery packets.
  - Eliminate redundant broadcasts.
- Write an Instant Messenger application that runs on miniroute.
  - Read input from the user (look at read*).
  - Add miniterm_initialize to your system initialize functions.
  - miniterm_read will let you read from the keyboard.
Set `BCAST_ENABLED` to 1.

Set `BCAST_ADDRESS`:
- 192.168.1.255 for ad-hoc network (Google for instructions on how to setup an ad-hoc network).
- x.y.z.255 for CSUG lab.

When debugging:
- Set `BCAST_TOPOLOGY_FILE` to 1.
- Provide a topology file (see project description).
  - Test without wireless.
- Use only in CSUG lab.
Feeling ambitious?

Remove the routing cache timeout.

- Instead, detect broken links and re-perform discovery.
- Requires verifying that hops work.
- Take advantage of broadcasting to see when the next host forwards the packet.
  - Faster (less sends).
  - Requires more work.
Localized route patching.

- When hop-to-hop communication fails, have the hop that detects the failure perform a new route discovery.
- Patch the route on the failed packet to allow it to route successfully.
- Source and destination should be updated to reflect new route.
Feeling even more ambitious?

Cache aggressively.

- There are lots of opportunities to cache more.
- Every packet presents the chance to update the cache.
  - Some data is not worth caching.
Feeling *even more* ambitious?

- Keep redundant routes in cache.
- Keep multiple routes to the same destination in cache.
- When the source receives an error, the backup route may be used.
If you eat and breathe this stuff

- Hybrid proactive/reactive routing protocols
- See Professor Sirer’s SHARP‡

Concluding thoughts

- Have some fun with this project.
- It’s much less work than P4, and much more fun too.
- Come see the TAs in office hours.

§some of the TAs get lonely
Ad-Hoc Networking

Ayush Dubey
dubey@cs
November 2, 2012