What is an overlay network?

- A network whose links are based on (end-to-end) IP hops
- And that has multi-hop forwarding
  - I.e. a path through the network will traverse multiple IP “links”
- As a “network” service (i.e. not part of an application per se)
  - I.e. we are not including Netnews, IRC, or caching CDNs in our definition
Why do we have overlay networks?

Historically (late 80’s, early 90’s):
- to get functionality that IP doesn’t provide
- as a way of transitioning a new technology into the router infrastructure
  - mbone: IP multicast overlay
  - 6bone: IPv6 overlay

More recently (mid-90’s):
- to overcome scaling and other limitations of “infrastructure-based” networks (overlay or otherwise)
- Yoid and End-system Multicast
  - Two “end-system” or (nowadays) “peer-to-peer” multicast overlays
- Customer-based VPNs are kind-of overlay networks
  - If they do multi-hop routing, which most probably don’t
Why do we have overlay networks?

- Still more recently (late-90’s, early 00’s):
  - to improve the performance and reliability of native IP!!
- RON (Resilient Overlay Network)
  - Work from MIT (Andersen, Balakrishnan)
  - Based on results of Detour measurements of Savage et.al., Univ of Washington, Seattle

End-to-end effects of Internet Path Selection

- Savage et. al., Univ of Washington Seattle
- Compared path found by internet routing with alternates
  - Alternates composed by gluing together two internet-routed paths
- Roundtrip time, loss rate, bandwidth
- Data sets: Paxson, plus new ones from UW
- Found improvements in all metrics with these “Detour” routes
BGP and other studies
- Paxson (ACIR), Labovitz (U Mich), Chandra (U Texas)
- Show that outages are frequent (>1%) 
- BGP can take minutes to recover

RON Rational
- BGP cannot respond to congestion
- Because of information hiding and policy, BGP cannot always find best path
  - Private peering links often cannot be discovered
- BGP cannot respond quickly to link failures
- However, a small dynamic overlay network can overcome these limitations
BGP lack of response to congestion

- Very hard for routing algorithms to respond to congestion (route around it)
- Problem is oscillations
  - Traffic moved from congested link to lightly-loaded link, then lightly-load link becomes congestions, etc.
- ARPANET (~70 node network) struggled with this for years
  - Khanna and Zinky finally solved this (SIGCOMM ’89)
  - Heavy damping of responsiveness

BGP information hiding

Private peering link. Site1 and Site2 can exchange traffic, but Site2 cannot receive internet traffic via ISP1 (even if policy allows it).
BGP information hiding

Internet
ISP1  ISP2
30.1/16 20.1/16
30.1.3/24 X
20.1.5/24

Site1
Site2

Internet
ISP1  ISP2
30.1/16 20.1/16
30.1.3/24 X
20.1.5/24

Site1
Site2

RON can bypass BGP information hiding

Internet
ISP1  ISP2
30.1/16 20.1/16
30.1.3/24 X
20.1.5/24

Site1
Site2

RON1
30.1.3.5

RON2
20.1.5.7

RON3
RON test network had private peering links

BGP link failure response

- BGP cannot respond quickly to changes in AS path
  - Hold down to prevent flapping
  - Policy limitations
- But BGP can respond locally to link failures
  - And, local topology can be engineered for redundancy
Local router/link redundancy

- eBGP and/or iBGP can respond to peering failure without requiring an AS path change
- Intra-domain routing (i.e. OSPF) can respond to internal ISP failures

AS path responsiveness is not strictly necessary to build robust internets with BGP.

Note: the telephone signalling network (SS7, a data network) is built this way.

Goals of RON

- Small group of hosts cooperate to find better-than-native-IP paths
  - ~50 hosts max, though working to improve
- Multiple criteria, application selectable per packet
  - Latency, loss rate, throughput
- Better reliability too
- Fast response to outages or performance changes
  - 10-20 seconds
- Policy routing
  - Avoid paths that violate the AUP (Acceptable Usage Policy) of the underlying IP network
- General-purpose library that many applications may use
  - C++
Some envisioned RON applications

- Multi-media conference
- Customer-provided VPN
- High-performance ISP

Basic approach

- Small group of hosts
- All ping each other---a lot
  - Order every 10 seconds
  - 50 nodes produces 33 kbps of traffic per node!
- Run a simplified link-state algorithm over the $N^2$ mesh to find best paths
  - Metric and policy based
- Route over best path with specialized metric- and policy-tagged header
  - Use hysteresis to prevent route flapping
**Major results** (tested with 12 and 16 node RONs)

- Recover from most complete outages and all periods of sustained high loss rates of >30%
- 18 sec average to route around failures
- Routes around throughput failures, doubles throughput 5% of time
- 5% of time, reduced loss probability by >0.5
- Single-hop detour provides almost all the benefit

**RON Architecture**

Local or shared among nodes. Relational DB allows a rich set of query types.

Router is itself a RON client.
Conduit, Forwarder, and Router

**Ron Header** (inspired by IPv6!...but not IPv6)

---

**Runs under IP**

<table>
<thead>
<tr>
<th>Version</th>
<th>Hop Limit</th>
<th>Routing Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>RON Source Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RON Destination Address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source port | Dest port

Flow ID

Policy Tag

Packet Type

---

Runs over UDP

---

Performance metrics

IPv4

Unique per flow. Cached to speed up (3-phase) forwarding decision

Selects conduit
Link evaluation

- Defaults:
  - **Latency (sum of):**
    - \( \text{lat}_i \leftarrow \alpha \cdot \text{lat}_{i-1} + (1 - \alpha) \cdot \text{new\_sample}_i \)
    - exponential weighted moving average \((\alpha = 0.9)\)
  - **Loss rate (product of):**
    - average of last 100 samples
  - **Throughput (minimum of):**
    - Noisy, so look for at least 50% improvement
    - Use simple TCP throughput formula:
      \[ \frac{1.5}{(\text{rtt} \cdot \sqrt{p})} \]
      \( p \): loss probability
  - Plus, application can run its own

Responding to failure

- Probe interval: 12 seconds
- Probe timeout: 3 seconds
- Routing update interval: 14 seconds
RON overhead

<table>
<thead>
<tr>
<th>10 nodes</th>
<th>20 nodes</th>
<th>30 nodes</th>
<th>40 nodes</th>
<th>50 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8 Kbps</td>
<td>5.9 Kbps</td>
<td>12 Kbps</td>
<td>21 Kbps</td>
<td>32 Kbps</td>
</tr>
</tbody>
</table>

- Probe overhead: 69 bytes
- RON routing overhead: 60 + 20 (N-1)
- 50: allows recovery times between 12 and 25 s

Two policy mechanisms

- **Exclusive cliques**
  - Only member of clique can use link
  - Good for “Internet2” policy
    - No commercial endpoints went over Internet2 links
- **General policies**
  - BPF-like (Berkeley Packet Filter) packet matcher and list of denied links
  - Note: in spite of this, AUPs may easily, even intentionally, be violated
RON deployment (19 sites)

To vu.nl lulea.se ucl.uk

To kaist.kr, .ve

.com (ca), .com (ca), dsl (or), cci (ut), aros (ut), utah.edu, .com (tx),
cmu (pa), dsl (nc), nyu, cornell, cable (ma), cisco (ma), mit,
vu.nl, lulea.se, ucl.uk, kaist.kr, univ-in-venezuela

AS view
Latency CDF

- ~20% improvement?
- RON improves latency by tens to hundreds of ms on some slower paths
- RON overhead increases latency by about 1/2 ms on already fast paths

Same latency data, but as scatterplot

- Banding due to different host pairs
RON greatly improves loss-rate

**30-min average loss rate with RON**

RON loss rate never more than 30%

13,000 samples

An order-of-magnitude fewer failures

**30-minute average loss rates**

<table>
<thead>
<tr>
<th>Loss Rate</th>
<th>RON Better</th>
<th>No Change</th>
<th>RON Worse</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>479</td>
<td>57</td>
<td>47</td>
</tr>
<tr>
<td>20%</td>
<td>127</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>30%</td>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50%</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>80%</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100%</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Resilience Against DoS Attacks

Throughput Improvement
Some unanswered questions

- How much benefit comes from smaller or larger RONs?
  - Would a 4-node RON buy me much?
- Do results apply to other user communities?
  - Testbed consisted mainly of high-bandwidth users (3 home broadband)
  - Research networks may have more private peering than residential ISPs

Some concerns

- Modulo unanswered questions, clearly RON provides an astonishing benefit
- However . . .
Some concerns

- Is RON TCP-unfriendly?
  - RON path change looks like a non-slowstarted TCP connection
  - On the other hand, RON endpoints (TCP) would back off after failure

- Would large-scale RON usage result in route instabilities?
  - Small scale probably doesn’t because a few RONs are not enough to saturate a link
  - Note: internet stability is built on congestion avoidance within a stable path, not rerouting
Some concerns

- RON’s ambient overhead is significant
  - Lots of RONs would increase overall internet traffic, lower performance
  - This is not TCP-friendly overhead
  - 32 Kbps (50-node RON) is equivalent to high-quality audio, low-quality video!
  - Clearly the internet can’t support much of this
  - RON folks are working to improve overhead

RON creators’ opinion on overhead

- “Not necessarily excessive”
- “Our opinion is that this overhead is not necessarily excessive. Many of the packets on today’s Internet are TCP acknowledgments, typically sent for every other TCP data segment. These “overhead” packets are necessary for reliability and congestion control; similarly, RON’s active probes may be viewed as “overhead” that help achieve rapid recovery from failures.”
Some concerns

- RONs break AUPs (Acceptable Usage Policies)
  - RON has its own policies, but requires user cooperation and diligence