Site Multihoming by IPv6 Intermediation (shim6)

Iljitsch van Beijnum

University of Cambridge Computer Lab, 2008-10-1

IPv6

- Addresses are now 128 rather than 32 bits
- Has some cool stuff like Neighbor Discovery
- But otherwise nothing new under the sun
- Especially for routing: mainly just like IPv4 routing with more bits

More bits

- So:
  - no artificial scarcity: everyone gets a /48
    - (well... maybe a /56, or...)
    - /48 = 65536 /64 subnets big enough for all ethernet cards ever built and more
  - what if millions of people want to multihome with that /48?

Multihoming

- Connect to two or more ISPs. Usually:
  - get "provider independent" addresses
  - announce these to the rest of the world through each ISP with BGP
  - if one ISP/link fails, packets are rerouted over another

Routing scalability

- Average packet size: ± 500 bytes
- So 2.5 million packets/s on 10 gigabit link
- 0.4 µs to do a routing table lookup
  - minimum size packets: 67 nanoseconds
- "Global routing table" now 269000 entries
- Works for now, but can't handle too much growth

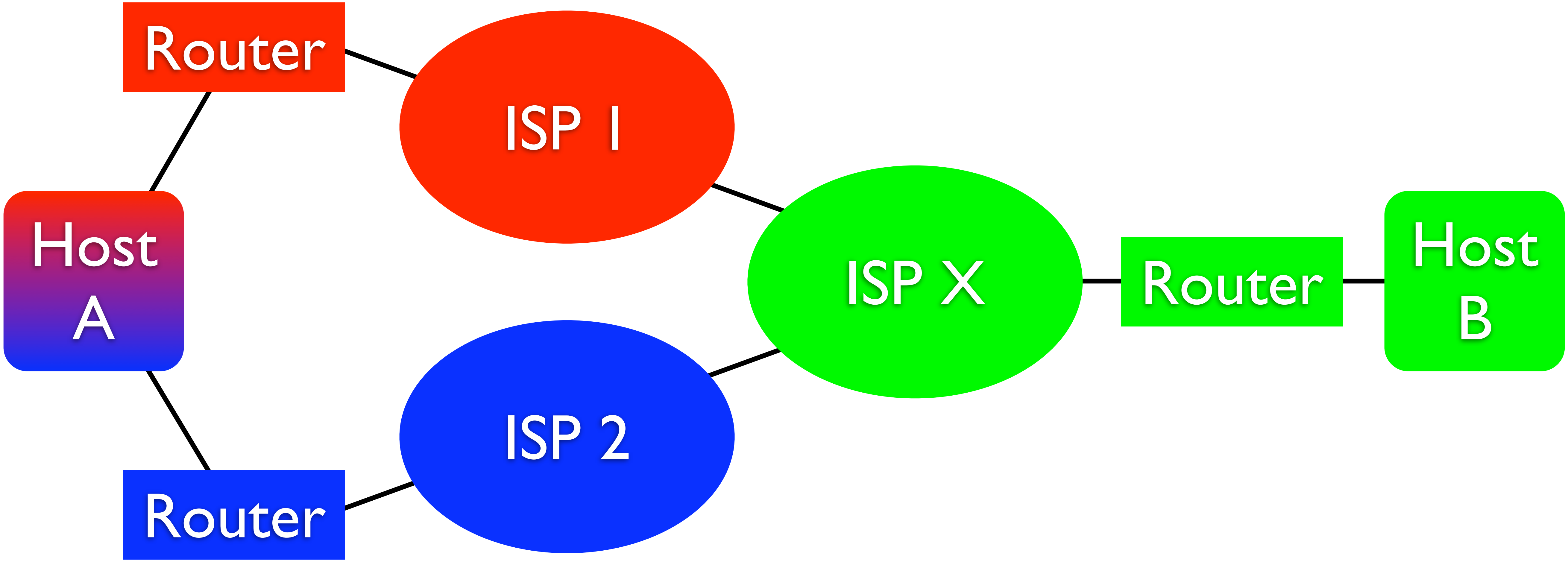
Scalable multihoming

- The routing system can't absorb millions of multihomers
- We need a solution!
- IETF multi6 wg tasked with that (~2001)
  - many proposals for solutions
  - several design teams

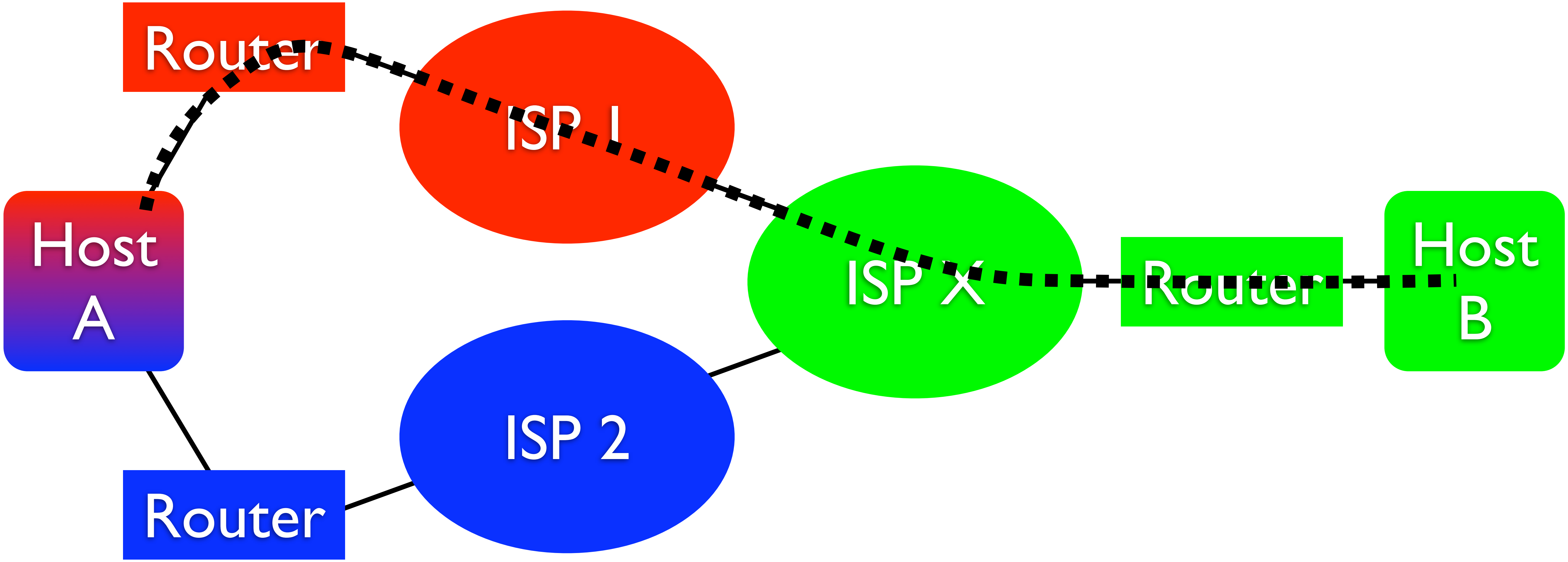
Outcome: shim6

- Host-based solution: each host does its own multihoming
- A host gets multiple addresses from multiple ISPs
- When (a link to) an ISP fails: switch addresses
- But hide address changes from "upper layer protocols" such as TCP/UDP

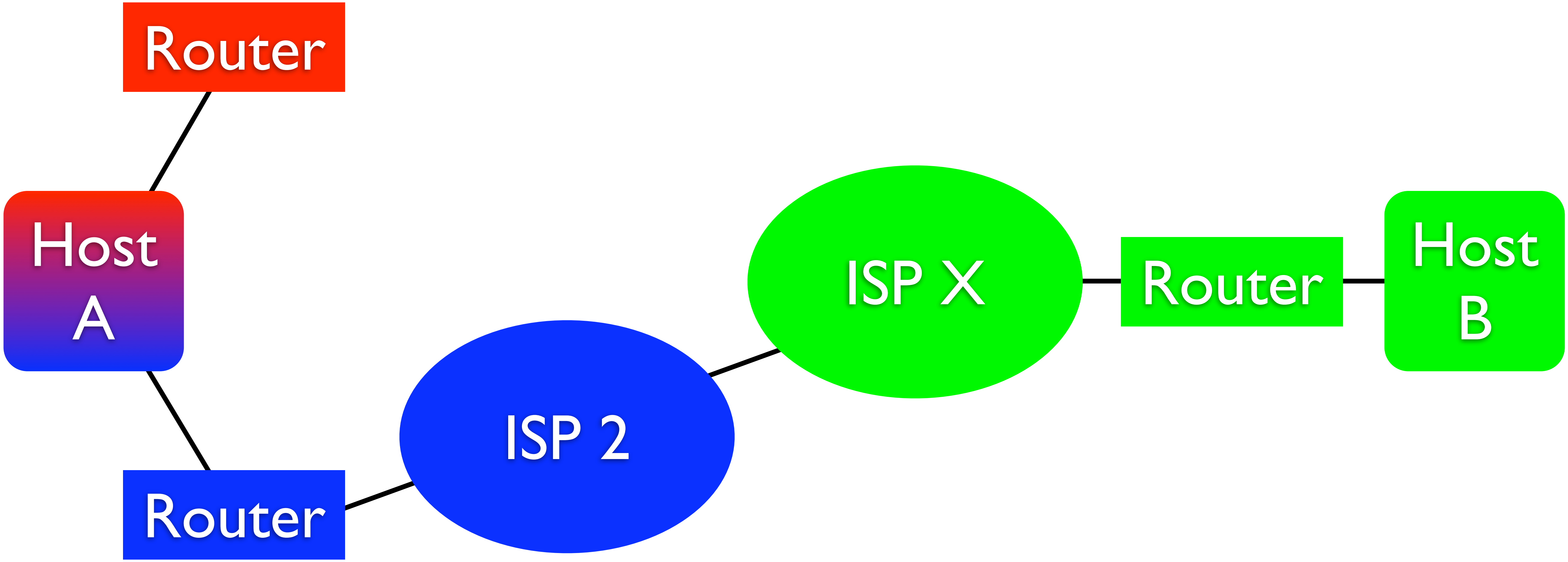
Shim6 operation



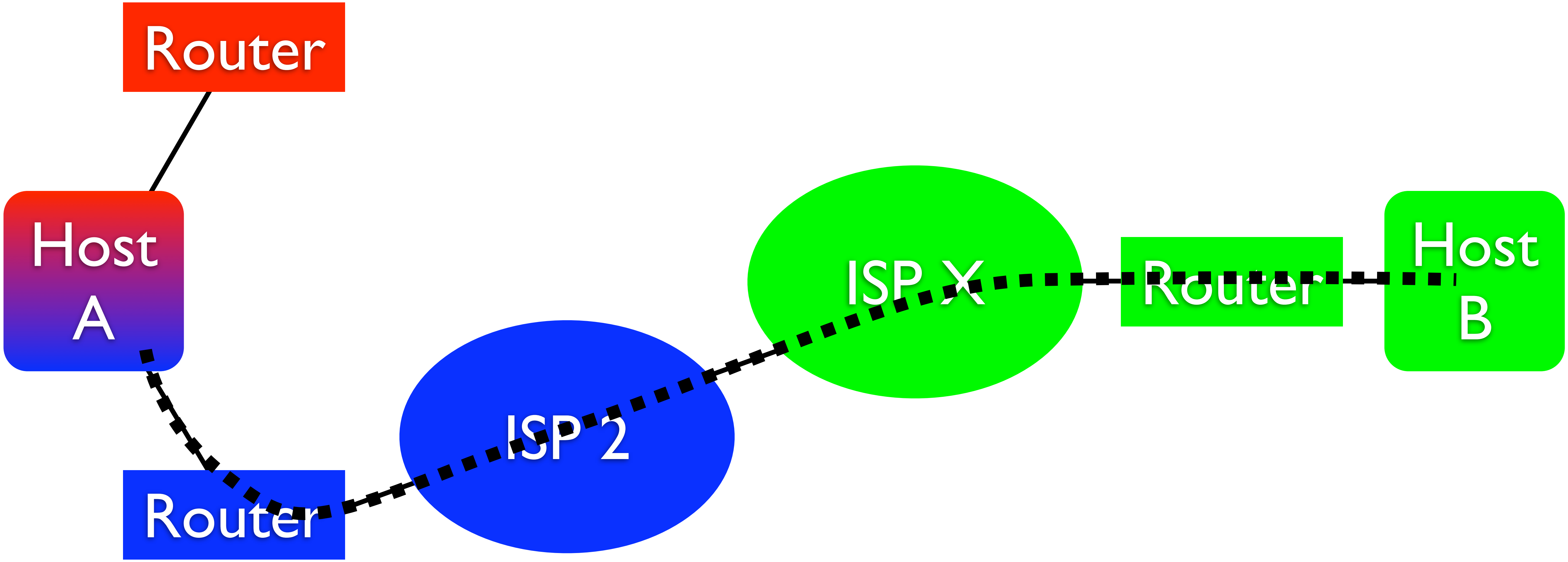
Shim6 operation



Shim6 operation



Shim6 operation



How?

1. Set up sessions as usual
2. After some packets, shim layer between IP and transports negotiates extra addresses
3. HBA for security
4. REAP for reachability detection
5. After failure, rewrite addresses and insert shim header

loc/id?

NEW NAMESPACE

- Is this a locator/identifier split?
  - sort of...
  - but the Upper Layer IDentifier (ULID) must also be a working locator address
  - so not really

loc/id?

NEW
NAMESPACE

- Is this a locator/identifier split?
  - sort of...
  - but the Upper Layer IDentifier (ULID) must also be a working locator address
  - so not really

Shim6 signaling

- Four-way handshake similar to HIP:
- `I1`: initiates, mostly nonces
- `R1`: reply with nonces to prevent DoS
- `I2`: just context state or also locators
- `R2`: confirms, may have locators

Security

- "Hi, my ID is windowsupdate.com, my locator is iljitsch.net!"
- Redirection attacks like this would be bad
- IPsec et al. not a solution: too heavy
- Shim6 uses Hash Based Addresses:
    - bottom 64 bits of IPv6 address contain a hash over all valid locators

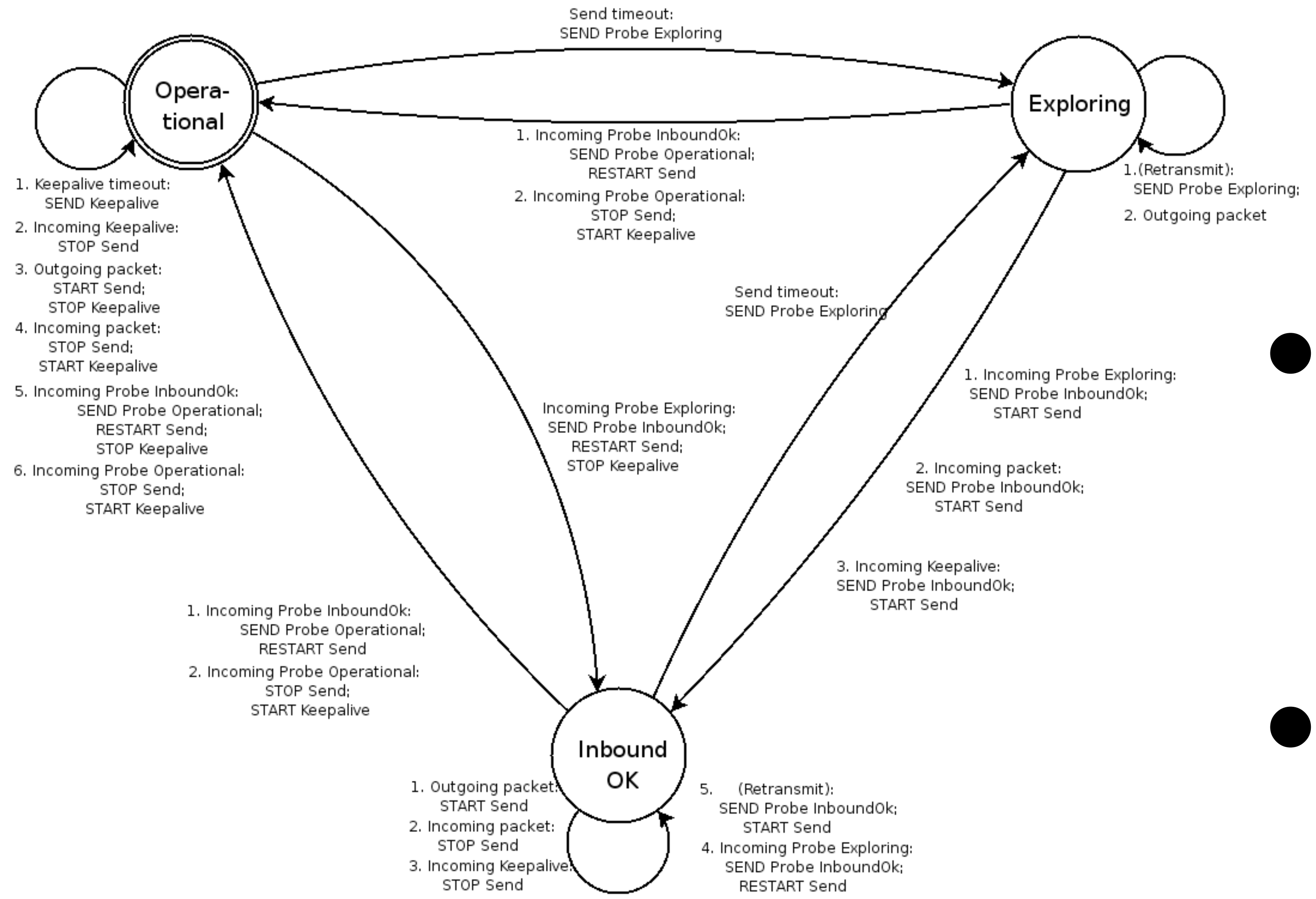
REAchability Protocol

- Assumption: traffic always flows in both directions
  - corollary: if we receive we *must* send. Shim will generate keepalives if needed
  - so if we send but don't receive: trouble
  - start exploration phase to find working addresses

REAP (2)

- Failures may be unidirectional
- So if one end initiates exploration phase, other enters it as well
- Send probes with increasing interval
- Echo back info from recent probe(s)
- If we see inbound packets, go into InboundOK

REAP (3)



Opera-
tional

Exploring

Inbound
OK

Send timeout:
SEND Probe Exploring

1. Incoming Probe InboundOk:
SEND Probe Operational;
RESTART Send
2. Incoming Probe Operational:
STOP Send;
START Keepalive

1. Keepalive timeout:
SEND Keepalive
2. Incoming Keepalive:
STOP Send
3. Outgoing packet:
START Send;
STOP Keepalive
4. Incoming packet:
STOP Send;
START Keepalive
5. Incoming Probe InboundOk:
SEND Probe Operational;
RESTART Send;
STOP Keepalive
6. Incoming Probe Operational:
STOP Send;
START Keepalive

1.(Retransmit):
SEND Probe Exploring;
2. Outgoing packet

Send timeout:
SEND Probe Exploring

Incoming Probe Exploring:
SEND Probe InboundOk;
RESTART Send;
STOP Keepalive

1. Incoming Probe Exploring:
SEND Probe InboundOk;
START Send
2. Incoming packet:
SEND Probe InboundOk;
START Send
3. Incoming Keepalive:
SEND Probe InboundOk;
START Send

1. Incoming Probe InboundOk:
SEND Probe Operational;
RESTART Send
2. Incoming Probe Operational:
STOP Send;
START Keepalive

1. Outgoing packet:
START Send
2. Incoming packet:
STOP Send
3. Incoming Keepalive:
STOP Send

5. (Retransmit):
SEND Probe InboundOk;
START Send
4. Incoming Probe Exploring:
SEND Probe InboundOk;
RESTART Send

- Continue until other end is also in InboundOk
- Then go to Operational and stop probing
- Start rewriting addresses into newly found ones

Shim6 header, rewriting

- No, this is not NAT: receiver restores addresses before handing packet to ULP
- Shim6 header is inserted for demultiplexing

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |     Length    |1|                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                             |
|                     Receiver Context Tag                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Shim6 issues

- Needs to be implemented on both sides!
- Need to update all hosts to make a site multihomed
- Can't repair non-working ULID
  - so apps must cycle through all addresses
- Need to renumber when switching ISPs
- No traffic engineering (yet)

More issues/status

- Pulls the rug from under certain apps
- Interaction with mobility, IPsec, SCTP
- Only works with IPv6...
- PI addresses now available for IPv6, so why bother implementing shim6?
- Documents almost ready for publication as RFC but not much progress this year
- One or two experimental implementations

My current work

- Multipath TCP: split a TCP session into subflows, send those over separate paths
  - good for resource pooling, fast reaction to outages/congestion
- Two approaches: change both ends and negotiate addresses per path, or only change sender

One-ended TCPm

- Only changing sender easier to deploy
- Fewer sequence number space issues
- Do per-path congestion control with SACK
- Selecting paths could be an issue:
  - get help from routers
  - or use shim6 to negotiate addresses

Questions?

- If you think of more later: iljitsch@bgpexpert.com