ILNPv6 in FreeBSD

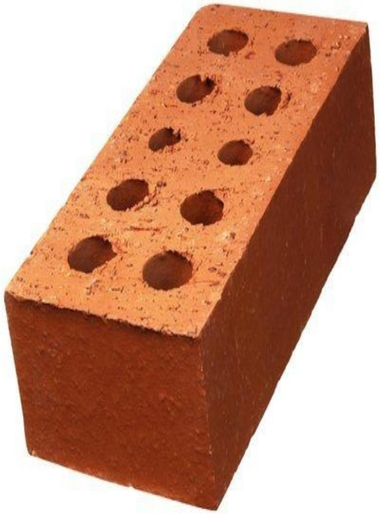
EuroBSDCon 2012 DevSummit

Bruce Simpson
School of Computer Science
University of St Andrews

With Saleem Bhatti
Sponsored by Cisco Systems, Inc.

Internet (With Fries)

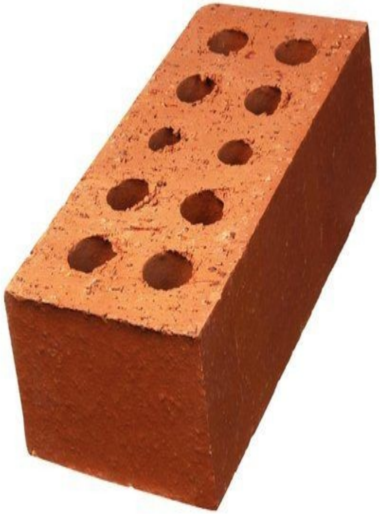
- The middleboxes are multiplying.
  - ILNP is middlebox birth control.
- Consider this list of features.
  - Host & Site Multi-homing
  - Host & Network Mobility
  - Multi-path capable transports: MP-TCP
  - Localised addressing: NAT
  - Traffic Engineering
  - Packet-level, end-to-end security.



- How do we solve these problems in one hit?

ILNP Boot Camp

- ILNP in plainer English:
  - Just cut IPv6 addresses in half
  - Bind sockets to the bottom half only
  - Exchange information about top half between endpoints
  - *Completely transparent to IPv6 core routers*



- Take two middleboxes into the shower?
  - No, just pf scrub and go

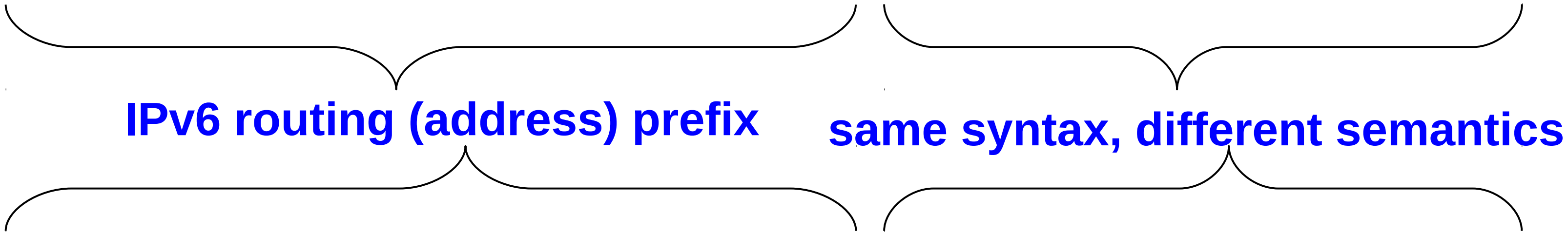
ILNPv6 Datagrams

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |   Next Hdr    |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        Source Locator                         +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                      Source Identifier                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                      Destination Locator                      +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                    Destination Identifier                     +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 addresses & ILNPv6

```
IPv6 (as in RFC3587 with RFC3177 – RFC6177 has 8-bit subnet ID):

 | 3 |     45 bits         |  16 bits   |       64 bits               |
 +---+---------------------+------------+-----------------------------+
 |001|global routing prefix| subnet ID  |    Interface Identifier     |
 +---+---------------------+------------+-----------------------------+
```



IPv6 routing (address) prefix

same syntax, different semantics

```
ILNPv6:

 |             64 bits                  |       64 bits               |
 +---+---------------------+------------+-----------------------------+
 |             Locator                  |     Node Identifier         |
 +---+---------------------+------------+-----------------------------+
```

same syntax and semantics as
IPv6 routing (address) prefix
so IPv6 core routers work as today

these bits only examined and
acted upon by end systems

Naming: IP vs. ILNP

Protocol Layer	IP	ILNP
Application	FQDN or IP address	FQDN (RFC1958)
Transport	IP address (+ port number)	Identifier (+ port number)
Network	IP address	Locator
(Interface)	IP address	(dynamic binding)

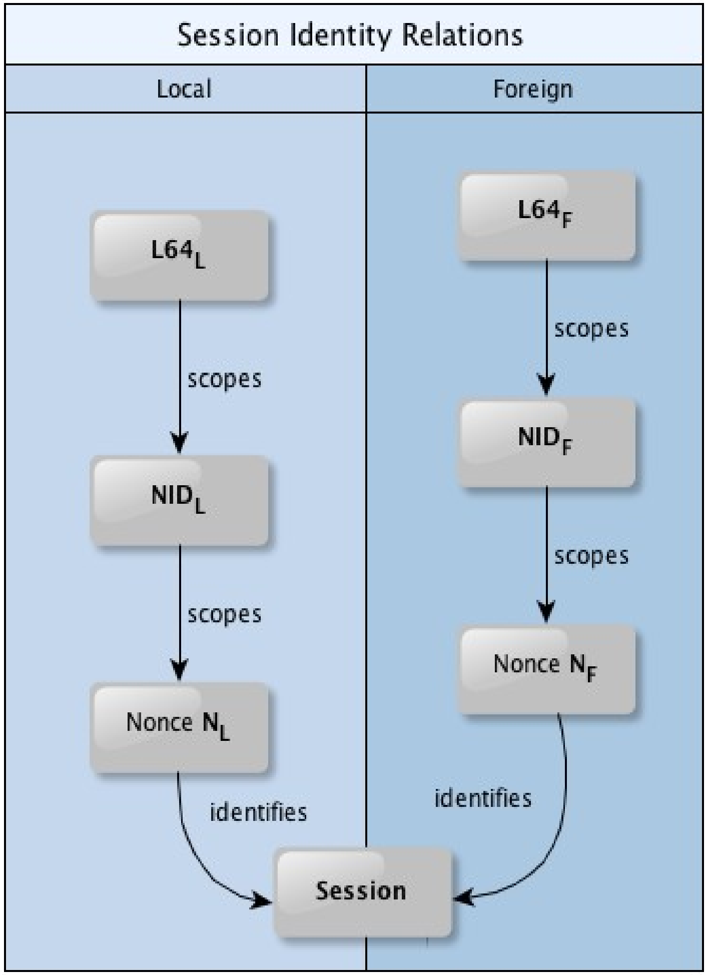
Entanglement **Separation**

FQDN = fully qualified domain name

ILNPv6 Locators & Identifiers

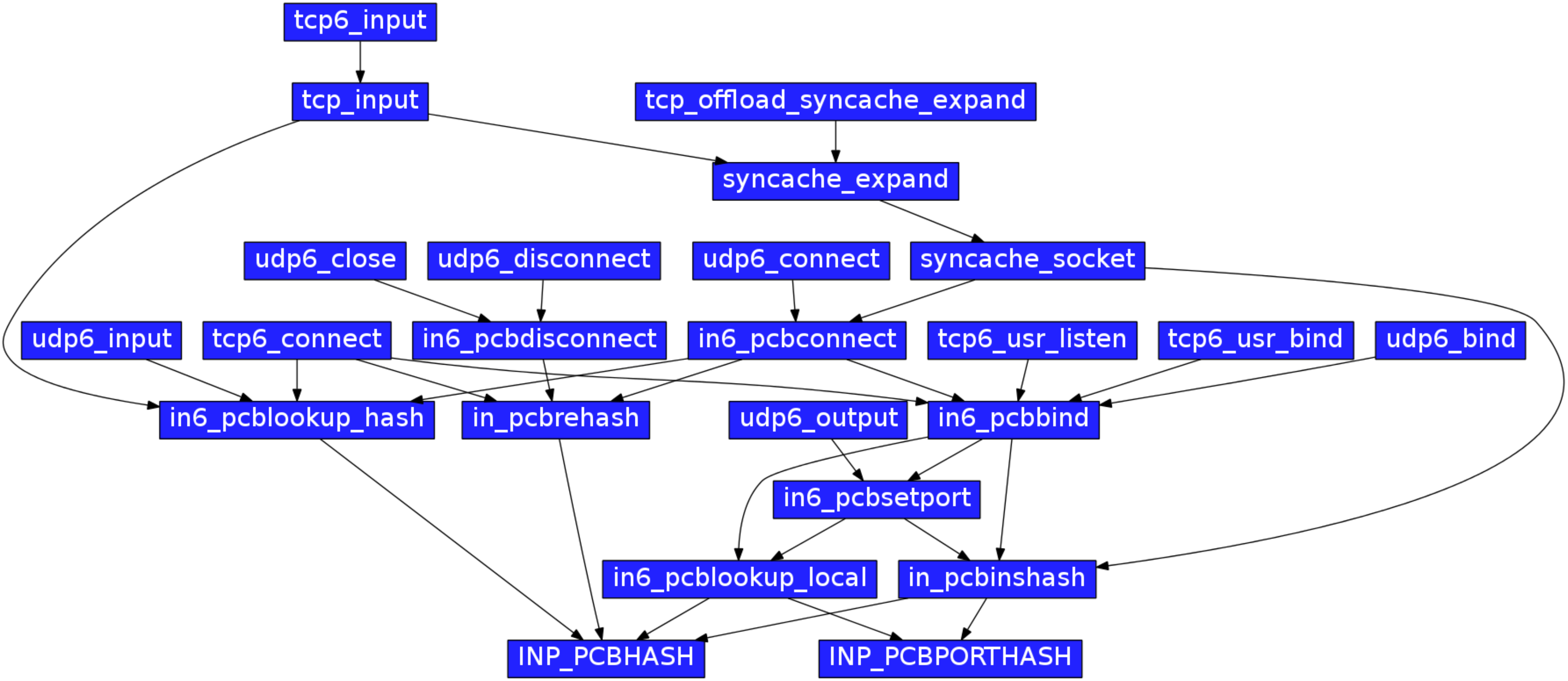
- ILNP functions manage:
  - Locator (L64) values
  - Node IDs (NIDs)
  - <L64, NID> bindings
- ILNP end-to-end state:
  - bound through NID values
  - <L64, NID> relation is 1:N

- Bindings are now **dynamic**



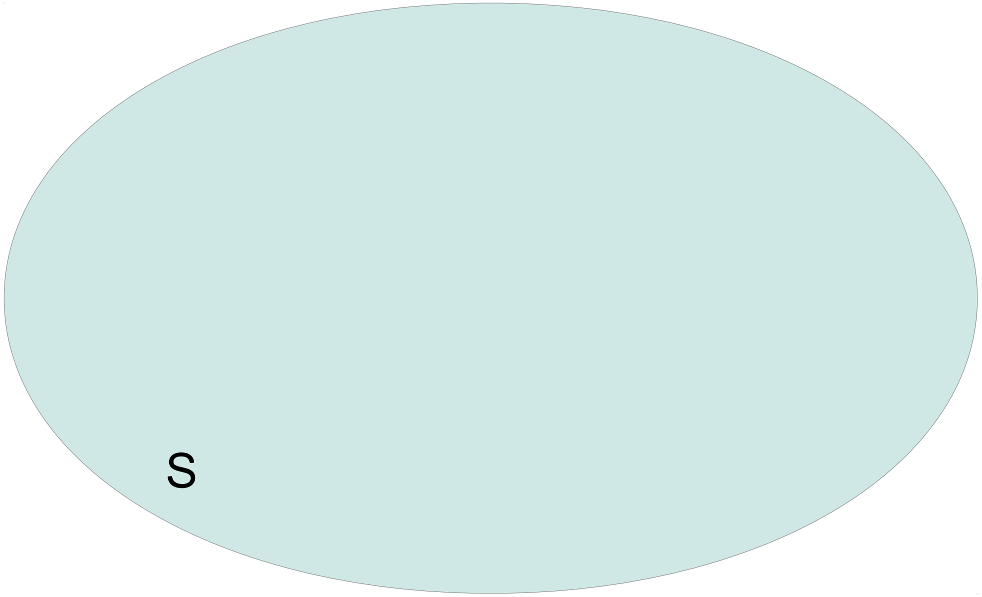
The binding problem

Where do we put ILNP?



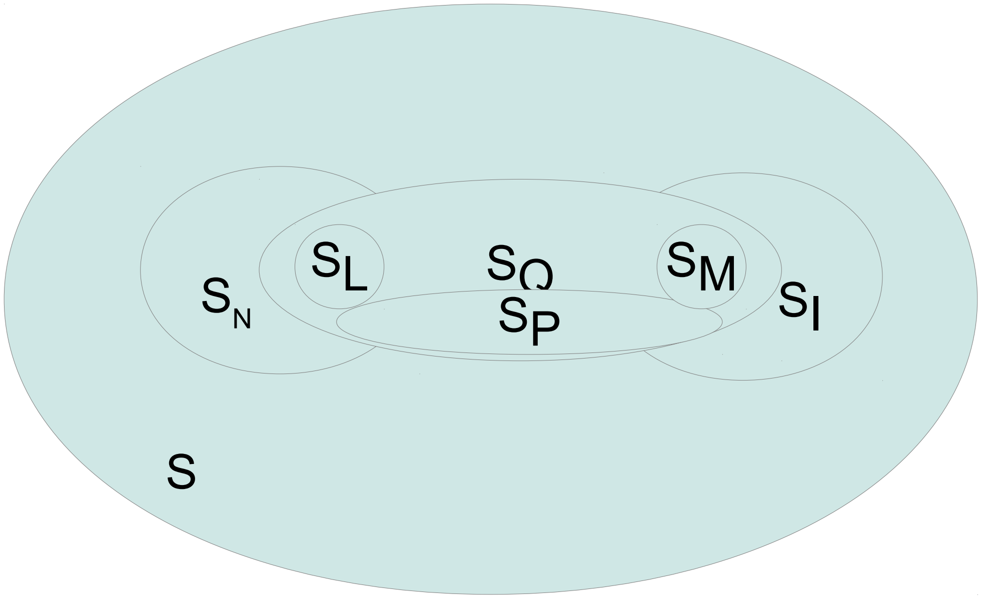
Sockets in IPv6

S All sockets



Sockets in dual-stack IPv6/ILNPv6

S	All sockets
S_I	ILNP sockets
S_N	Non-ILNP sockets
S_P	Sockets bound to local port NID
S_Q	Sockets bound to local port NID, wildcard
S_L	Non-ILNP sockets bound to local port NID
S_M	ILNP sockets bound to a NID



Dynamic Binding is harder than it looks

Current approach

- Network-layer ILNPv6, for ICMPv6 only
- Simulate ILNP DNS lookups in libc
- “Hobo Stove” Named Sockets
  - DNS, ILNP sessions coupled by *lookaside cache*
- Multi-homing and fail-over are controlled by the *locator selection procedure*
- Outcome: Changes are (mostly) transparent to legacy IPv6 apps

Implementation details

- Modifications to packet I/O in icmp6.c
- Simulate ILNP DNS lookups in libc
  - /etc/hosts syntax extended
  - RFC 3484 is bypassed in getaddrinfo(); not applicable to ILNP
- Poor man's named sockets
  - If a socket operation touches a recent lookup, ILNP session is activated.
- Locator selection uses existing mechanisms
  - We need a new in6_selectsrc()
  - For now, we just wrap it with a priority queue
- Outcome: 6000 line diff against 8.3-RELEASE to deliver Demo 2

Open questions

- What should an ILNPv6 API look like?
- How can we better support named sockets?
  - Coupling between DNS stub resolver and socket operations is required
- Long term: how will this cleanly interface with pf and pfsync?

Upcoming work

- Network and site mobility with **pf**
    - ILNP is not just for end-stations

- **pfSense** for interactive experiments

- Transport protocol support
    – TCP, UDP

- Not applicable to multicast or P2P (yet)

Thank you

- Questions? Comments?
  - Contact bms@ for demo
- Project web site

  `http://ilnp.cs.st-andrews.ac.uk/`

Bruce Simpson
School of Computer Science
University of St Andrews
http://www.cs.st-andrews.ac.uk/~bms/