In-Kernel End-System Multihoming with ILNP

University of
St Andrews

Gregor Haywood

EuroBSDCon September 2024

University of St Andrews

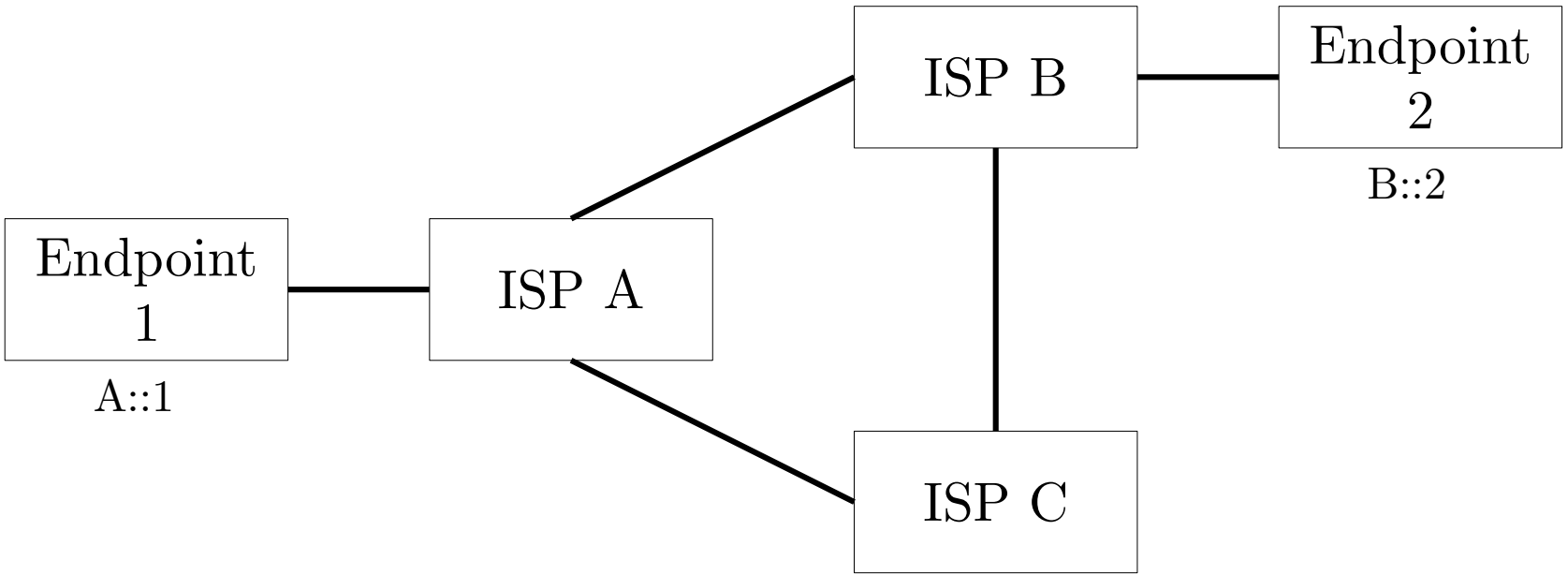
Overview

- The Multihoming Problem
- A Solution: ILNP
- FreeBSD Implementation
- Over-the-Internet Tests

The Multihoming Problem

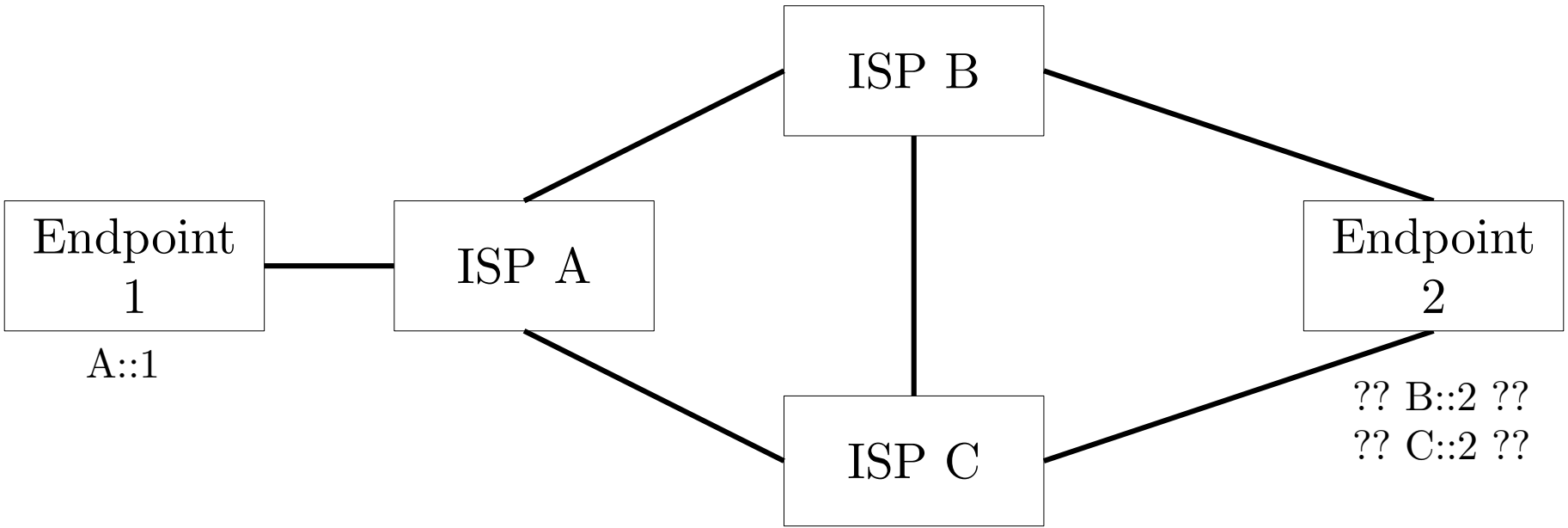
Once Upon a Time...

- One identity, one location, one address:



Now:

- One identity, **multiple** locations, ???

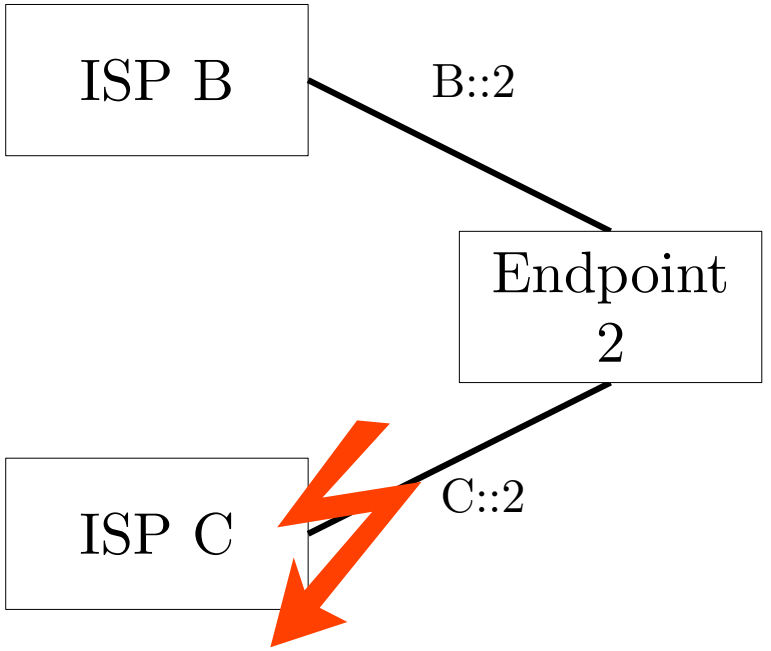


Address Semantics

- Addresses semantics are overloaded:
  - **Identify** a node
  - **Locate** a node
- Cannot change one without changing another!
  - Multihoming problem
  - (Mobility problem)

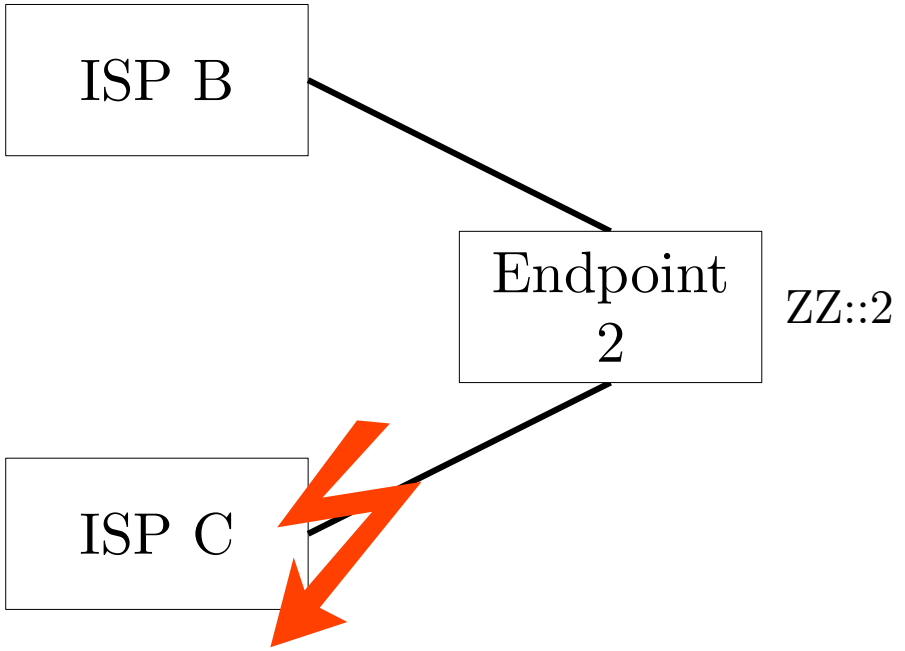
Multiple Addresses

- Transport sessions use B::2 or C::2
- No failover if one link goes down!
- No link aggregation
- No dynamic load balancing



One Address

- Transport sessions uses PI address ZZ::2
- Use any link dynamically
- PI addresses are bad:
  - Address space fragmentation
  - Cost



Fix Applications?

- Using multiple addresses **almost** works
- Can applications fix the problems?
  - API for adding new addresses
- QUIC, MP-TCP
- Protocol nimbyism?
- No one fix – must do every application

Use Middleboxes?

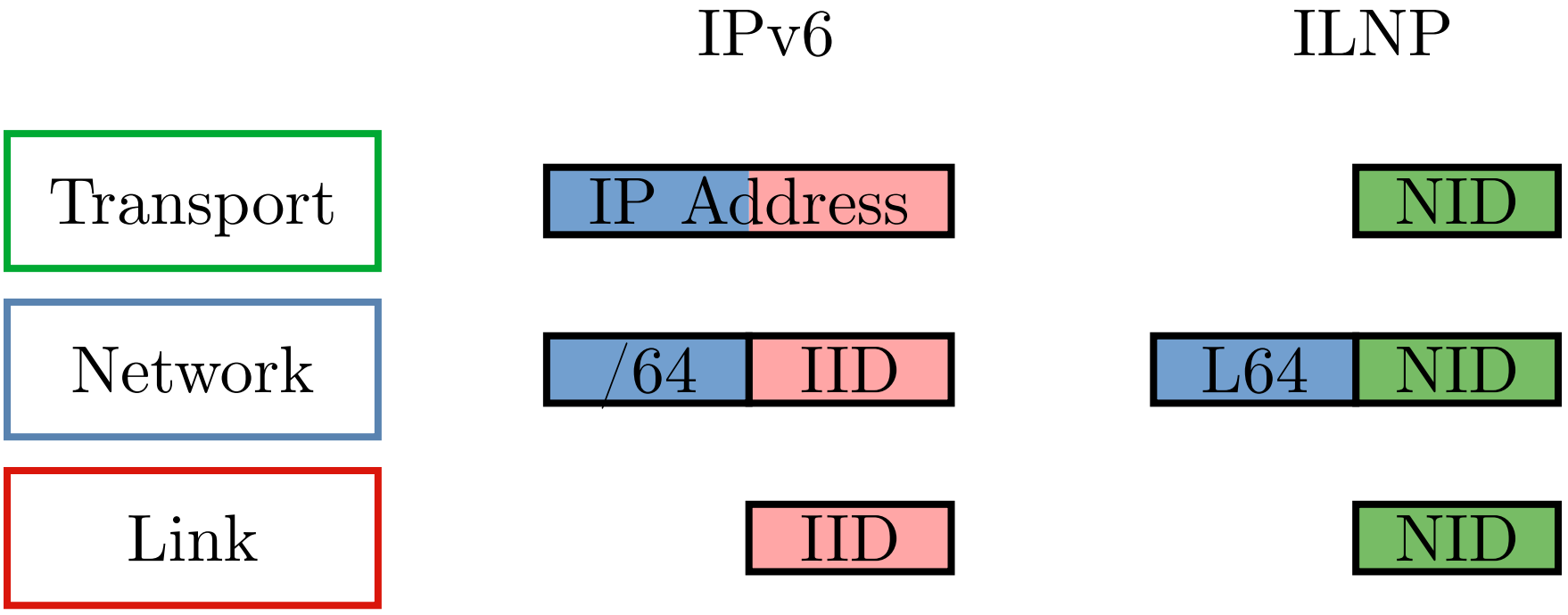
- Use tunnels/overlays/middleboxes
- Hide complexity from endpoint **and** the network
- Hidden complexity isn't a solution!
  - Breaks mobility mechanisms
  - May break future innovations

A Solution in Principle

- In-Kernel: fix once
- No New API: simplify applications
- End-System: keep complexity visible
- Multiple PA Routing Prefixes
- Single Transport ID

A Solution: ILNP

Architecture



NID Semantics

- Same syntax as IID
  - Managed via NDP
  - Can be random and ephemeral
- Identifies a whole node
- **Invariant** within a transport session

L64 Semantics

- Same syntax and semantics as a routing prefix
- **Excluded** from transport state
  - Checksums
  - PCB Lookups
- Dynamically bound to NIDs

Locator Updates

- New signalling needed to update set of active L64s
- Uses new ICMP messages
- Updates kernel data structures

Implementing ILNP

Overview

- Manage bindings in the ILCC (new data structure)
- TCP and UDP update bindings when sending
- ILCC updated by router advertisements and locator updates

UDP

- Very easy!
- Update PCB with latest L64 when sending
- Exclude the L64 from PCB lookups

```
#ifdef ILNP6
    /* Check if a connection uses ILNP6, and add the nonce. */
    if (inp->inp_flags2 & INP_ILNP6_CONNECTED)
        ilnp6_update_inpcb(&V_ilcc, inp, NULL);
#endif
```

PCB Lookups

- Hash the NID instead of the address
- Otherwise unchanged!

```
head = &pcbinfo->ipi_hash_exact[INP6_PCBHASH(&nid, lport, fport,
    pcbinfo->ipi_hashmask)];
CK_LIST_FOREACH(inp, head, inp_hash_exact) {
    /* Modified from in6_pcblookup_exact_match */
    if ((inp->inp_vflag & INP_IPV6) == 0)
        continue;
    if (ILNP6_NID_MATCH(&inp->in6p_faddr, faddr) &&
        ILNP6_NID_MATCH(&inp->in6p_laddr, laddr) &&
        inp->inp_fport == fport && inp->inp_lport == lport)
        return (inp);
}
```

ILNP Nonce Header

- ILNP packets include an extension header
- Extension headers already supported!
  - Trivial to implement this
  - No other use cases?

Session Initiation

- Modified libc initialises ILCC with remote node's NID and L64s
- Local NIDs set explicitly*
- Local L64s added by router advertisments
- PCB creation defaults to ILNP

*well, actually...

And the There's TCP...

TCP

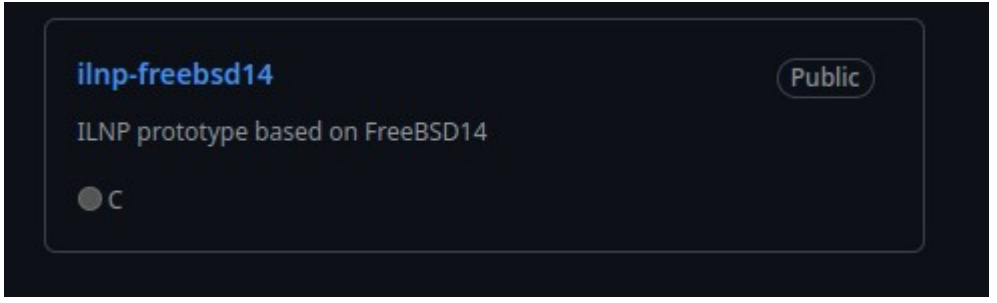
- In principle, just like UDP
- In *practice*, much trickier
  - Dual IPv6 and IPv4 support
  - SYN caching, so no PCB

Open Question: L64 Selection

- Current implementation uses round robin path selection
  - For privacy experiments, not performance
- Oblivious congestion control
- Retransmit on same L64? ACK on same L64?

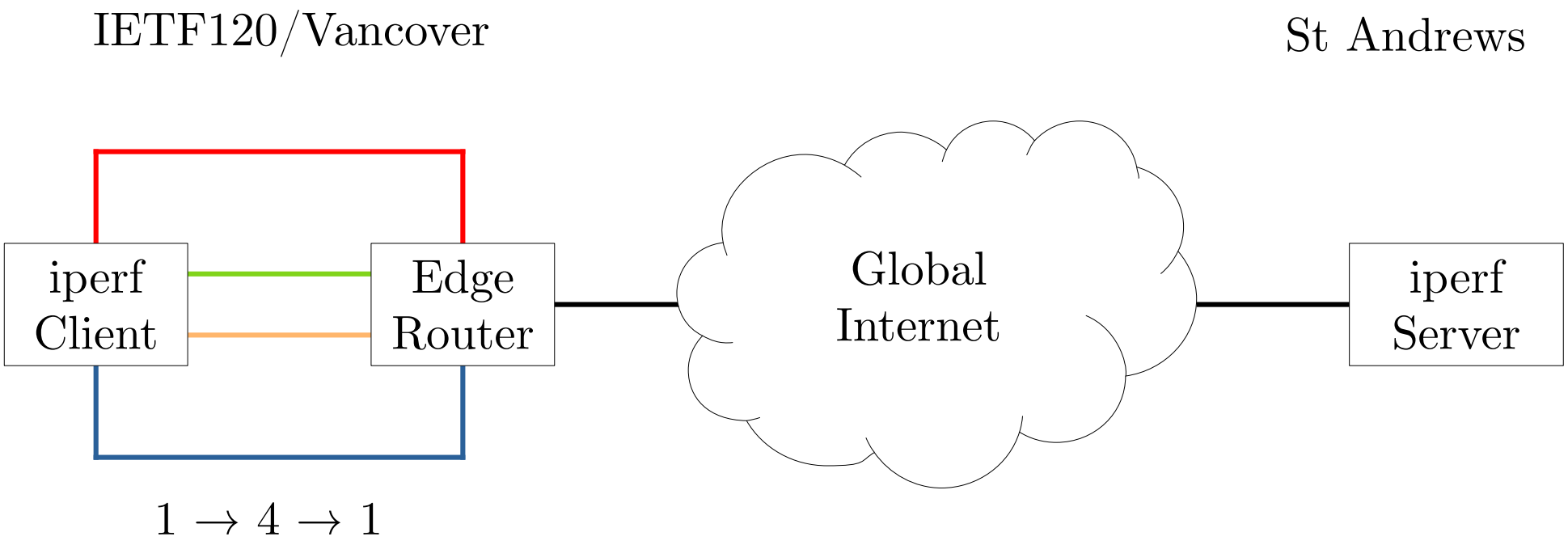
Code Release

- Working prototype
- Code is on GitHub
- Based on FreeBSD 14
- Development ongoing, with plans to eventually merge



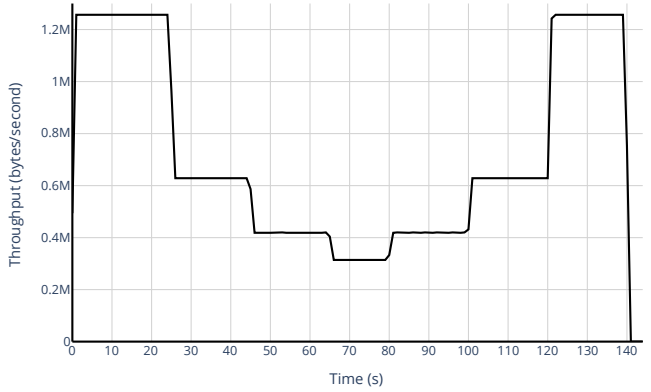
Round the World

The Setup

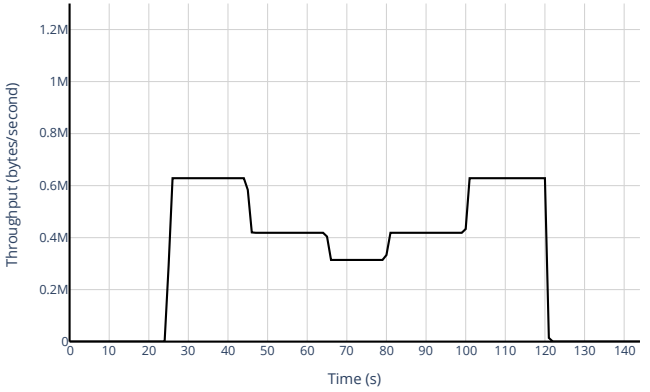


UDP: Per Interface

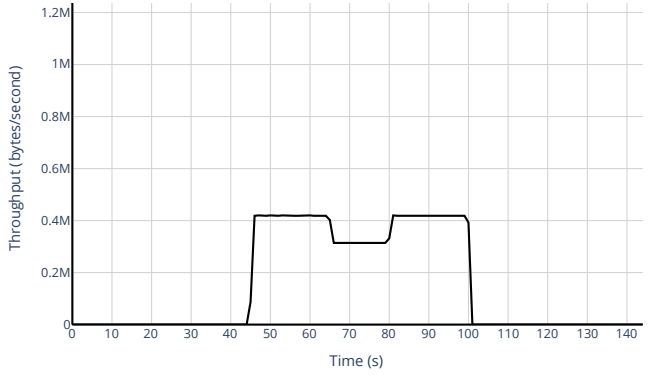
ix0



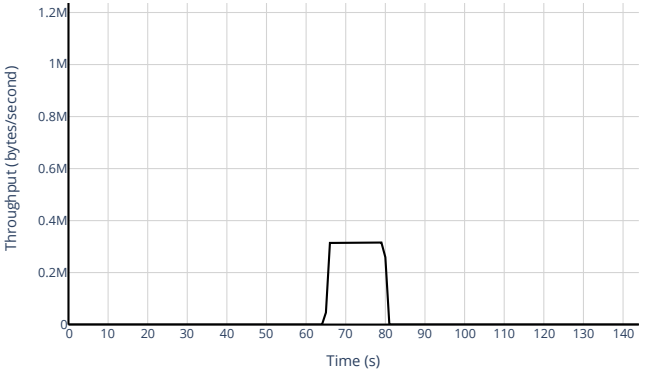
ix1



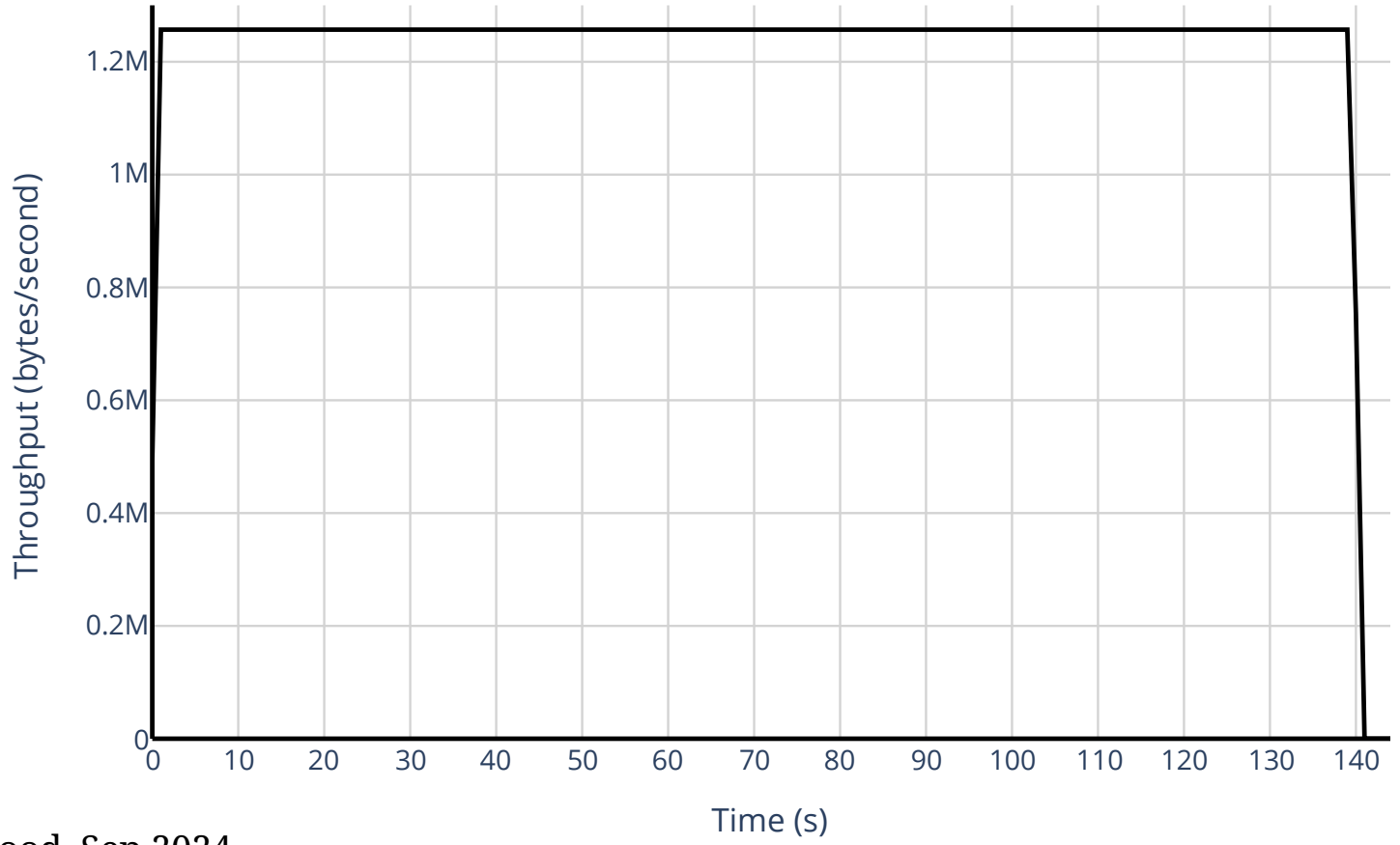
ix2



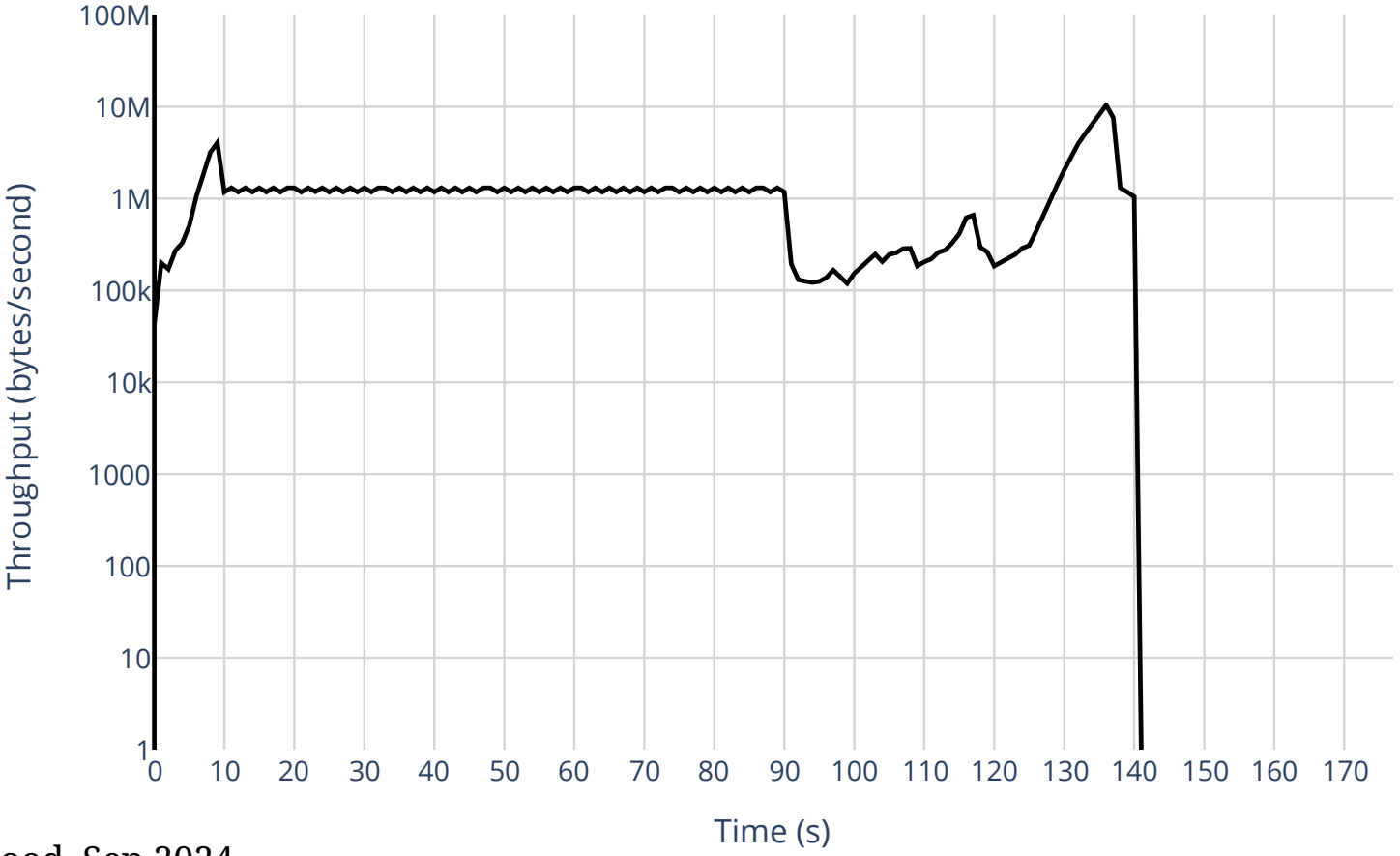
ix3



UDP: Total Throughput

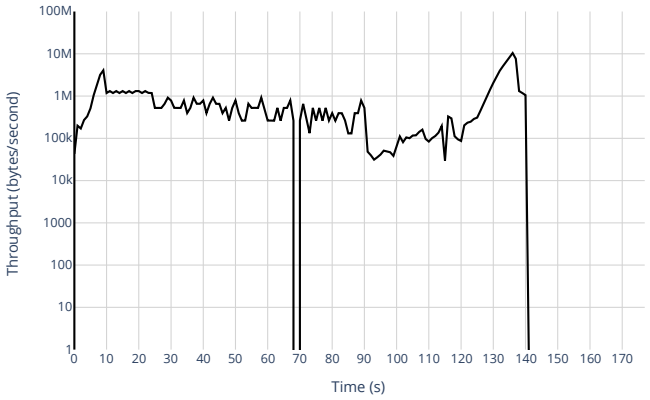


TCP: Total Throughput

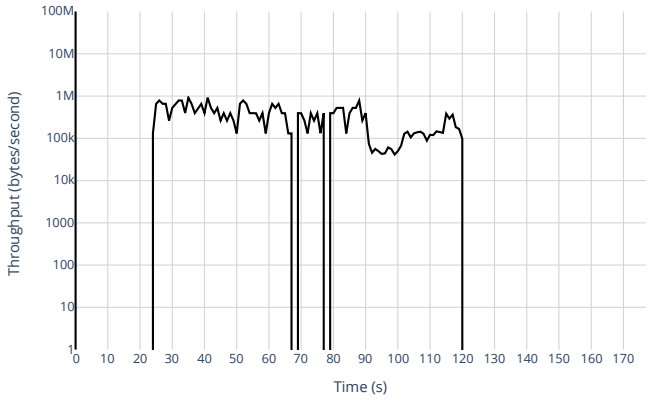


TCP: Per Interface

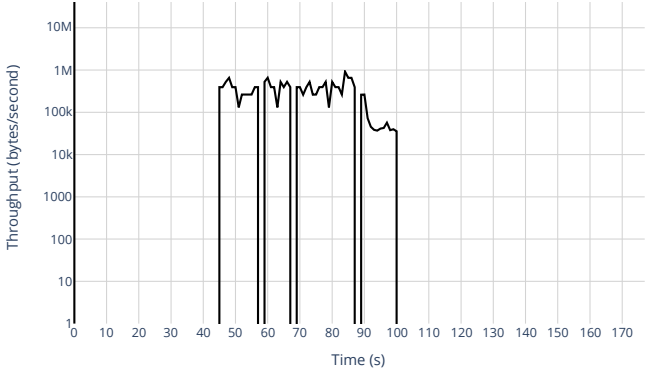
ix0



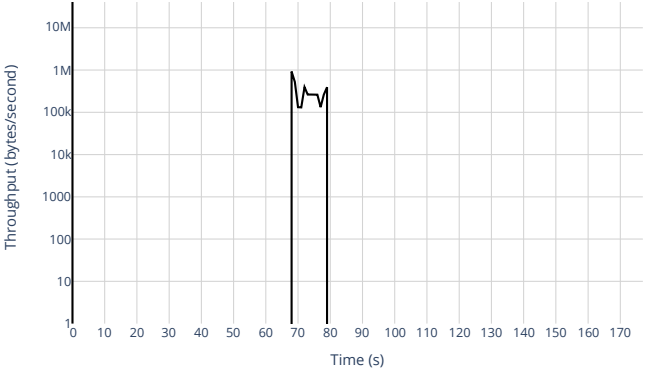
ix1



ix2



ix3



Key Takeaways

- End-system multihoming is possible!
- ILNP is publicly globally routable!
- “Fair” multipath utilisation with UDP
- Needs multipath-aware congestion control for TCP

Questions?

Code Release

gregorhaywood.com