University of St Andrews

# Are we stuck with the current Internet Protocol (IP)?

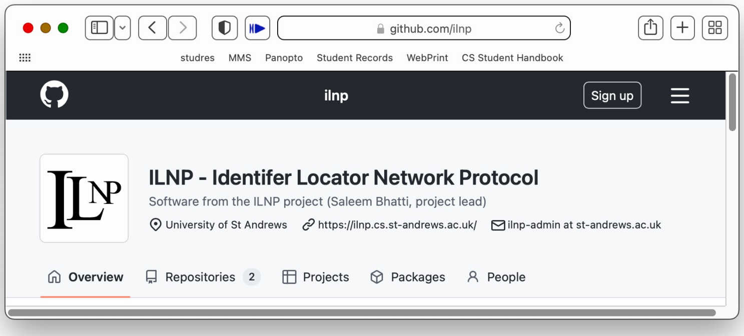
## And does it matter if we are?

Saleem Bhatti <saleem@st-andrews.ac.uk>
School of Computer Science, University of St Andrews.

University of St Andrews

# Ongoing research on Internet architecture

- ILNP: [https://ilnp.cs.st-andrews.ac.uk](https://ilnp.cs.st-andrews.ac.uk)
- Thanks to the work of many students at UStA ☺
- Results from the work of several students (in alphabetical order):
  - David Fergusson (control plane probe/measurement tool)
  - Gregor Haywood (FreeBSD 13)
  - Dr Ditchaphong Phoomikiatisak (Linux kernel v3.9)
  - Khawar Shehzad (Linux kernel v4.9 LTS, Verisign)
  - Bruce Simpson (FreeBSD 8, Cisco)
  - Ryo Yanagida (Linux kernel v4.9 LTS, Time Warner)
  - (Plus other students on sub-projects …)




- Discussions with colleagues, students, and friends over many years:
  - Academia, Industry, IETF/IRTF.

# Where are we with moving on from IPv4?

- We have "run out" of IPv4 addresses (no more to distribute).
- IPv6 delays:
  - Standardisation (after around 20 years [RFC8200]).
  - Deployment is still patchy (upgrade of equipment).
  - Overall usage is low.
- Did IPv6 solve any problems apart from address space?
  - IPv6 addresses are 128 bits compared to 32 bits for IPv4. ☺
  - But architectural principles for address usage remain broadly similar to IPv4.
- Very similar research challenges remain to improve IPv6 as for IPv4.

*[RFC8200] S. Deering, R. Hinden. "Internet Protocol, Version 6 (IPv6) Specification", RFC8200(S) / STD86, July 2017. https://datatracker.ietf.org/doc/html/rfc8200*

University of St Andrews

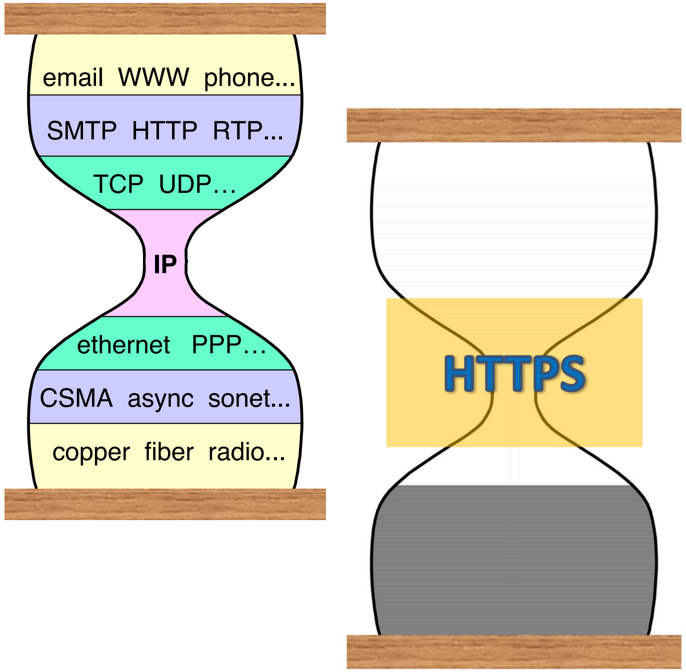
# Are we stuck with the current Internet Protocol (IP)?

Billions of people are using IP!

It all works just great!

Is there anything actually wrong with IP?

# The Interwebs




https://iab.org/wp-content/IAB-uploads/2011/03/hourglass-london-ietf.pdf

**Everything runs over HTTP! [PGS2010]**

**Well, not quite … but, web apps are popular …**

**Applications**:
- Wide-range of applications possible (esp. client-server).
- Flexible UIs possible (desktop, smartphone, tablet, etc).

**Development**:
- Well-defined APIs / SDKs / toolkits / frameworks.
- Javascript + libraries (lots of functionality).
- Relatively low barrier to entry for developers.

**Deployment**:
- Browser, or browser environment (webkit etc).
- Hosting for servers and services.
- No problems with firewalls.

*[PGS2010] L. Popa, A. Ghodsi, I. Stoica. 2010. HTTP as the narrow waist of the future internet. Proc. 8th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX). ACM. Article 6, 6 pages.* https://doi.org/10.1145/1868447.1868453

University of St Andrews

# Unintended consequences of Internet success

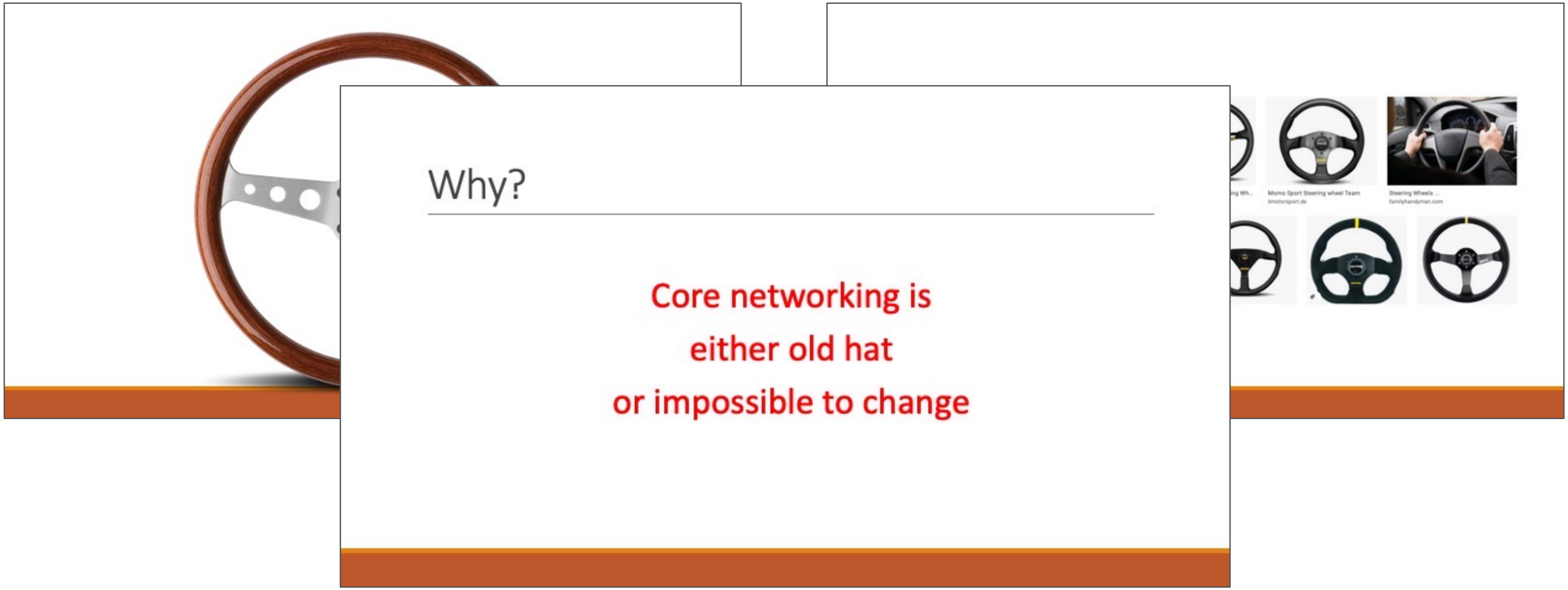
- Things nobody foresaw for the Internet [OB2018]:
  - (though [Postman1987] is a good holiday read! )
- Centralisation of ownership threatens utility ☹ :
  - "Wildly successful" applications means commercial (self-)interest can dominate global actions.
- Governance and control for benefit of citizens ☹ :
  - Large, well-known commercial actors vs. diversity of many smaller actors?
- Networking research and development trends 😐 :
  - Commercial impetus leads to narrow, commercial focus?
- **Ossification** of the infrastructure ☹:
  - **Hard to change infrastructure, constrains innovation.**



*[OB2018] N. Oever, D. Beraldo. Routes to rights: Internet architecture and values in times of ossification and commercialization. XRDS: Crossroads, The ACM Magazine for Students, 4 (July 2018), pp28-31.* https://doi.org/10.1145/3220561

*[Postman1987] Neil Postman. Amusing Ourselves to Death. Feb 1987. ISBN-13 : 978-0413404404*

# Polishing our steering wheels




“3 Futures for Computer Networking Research”, S. Keshav, Keynote talk, TMA2021, 14-15 Sep 2021, virtual/online

University of St Andrews

# Looking backwards to move forwards

# Research challenges – all retrofits to IPv4

- (Long standing research challenges [RFC3869])
- **Naming (in the general sense, but especially in relation to IP addresses).**
- Mobility:
  - individual nodes & whole networks.
- Multipath connectivity and Multihoming:
  - individual nodes & whole networks.
  - multipath transport protocols.
- End-to-end security and privacy (network packet level).
- (Others challenges also …)
- Currently, IP has independently designed **solutions** for such functionality, each with modified address usage that is not directly co-compatible.
  - Is **harmonised** functionality possible: any/all of these things together**?**

*[RFC3869] R. Atkinson (Ed), S. Floyd (Ed). "IAB Concerns and Recommendations Regarding Internet Research and Evolution", RFC3869(I), Aug 2004.* https://datatracker.ietf.org/doc/html/rfc3869

University of St Andrews

# IPv6 makes things (a little) better: examples

- "IPv6 Node Requirements" [RFC8504]
- **Naming: Larger addresses (128 bits) compared to IPv4 (32 bits).**
- Mobility:
  - Mobile IPv6 has better control plane compared to Mobile IPv4 [RFC6275].
  - MAY be implemented but is not REQUIRED.
- Multihoming (multiple connectivity, e.g. multiple ISPs):
  - If a node is multihomed, then follow [RFC8028].
- Security and Privacy:
  - Security: IPsec SHOULD be used but is not REQUIRED.
  - Privacy mechanism for address values SHOULD be used but is not REQUIRED.
- (Also others improvements … but IPv6 has its own, new problems, also.)

*[RFC6275] C.Perkins (Ed), D Johnson, J. Arkko. "Mobility Support in IPv6", RFC6275(PS), Jul 2011. https://datatracker.ietf.org/doc/html/rfc6275*
*[RFC8028] F. Baker, B. Carpenter. "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC8028(PS), Nov 2016. https://datatracker.ietf.org/doc/html/rfc8028*
*[RFC8504] T. Chown, J. Loughney. T. Winters. "IPv6 Node Requirements", RFC8504(BCP) / BCP220, Jan 2019. https://datatracker.ietf.org/doc/html/rfc8504*

University of St Andrews

# Naming and IP addresses

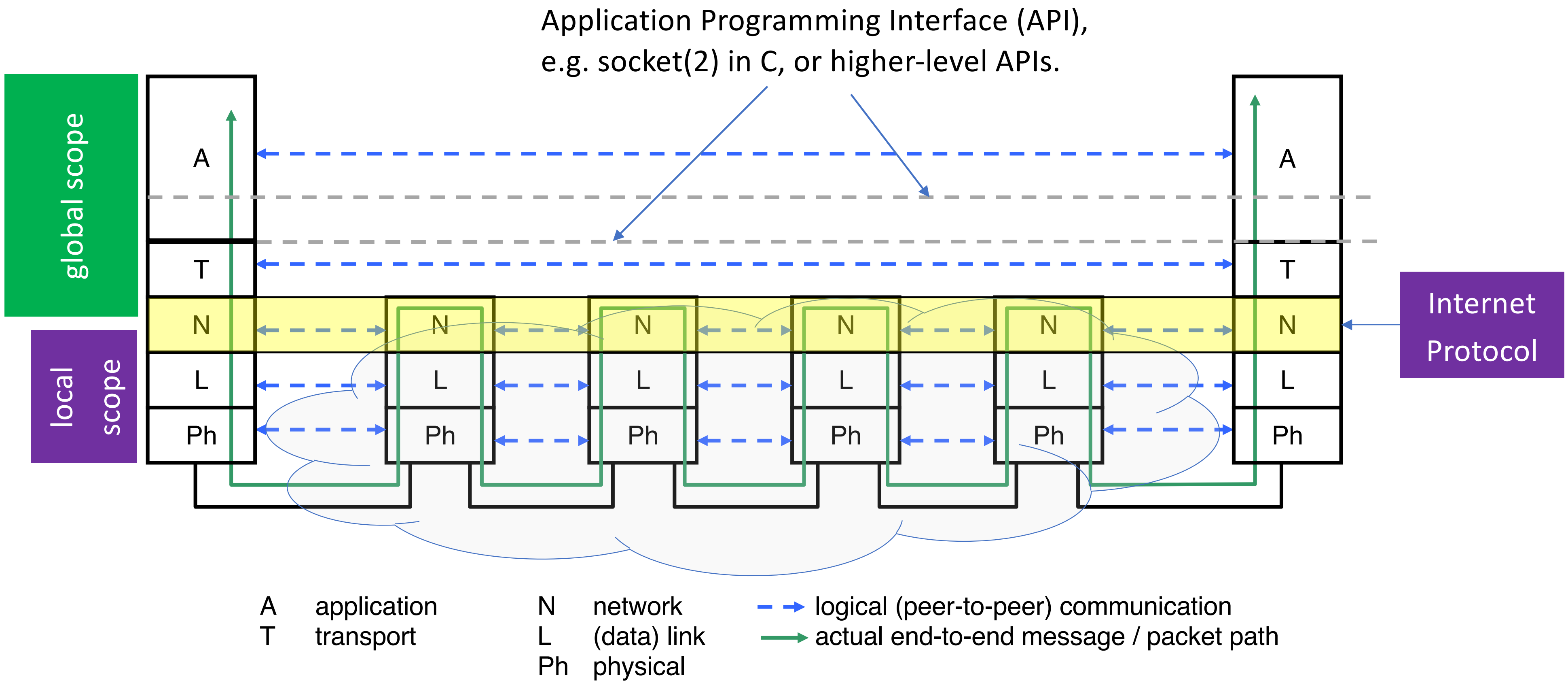
**"IP addresses considered harmful".**
Brian E. Carpenter.
ACM SIGCOMM CCR, vol. 44, issue 2, Apr 2014
https://doi.org/10.1145/2602204.2602215


**Abstract**
**This note describes how the Internet has got itself into deep trouble by over-reliance on IP addresses and discusses some possible ways forward.**

University of St Andrews

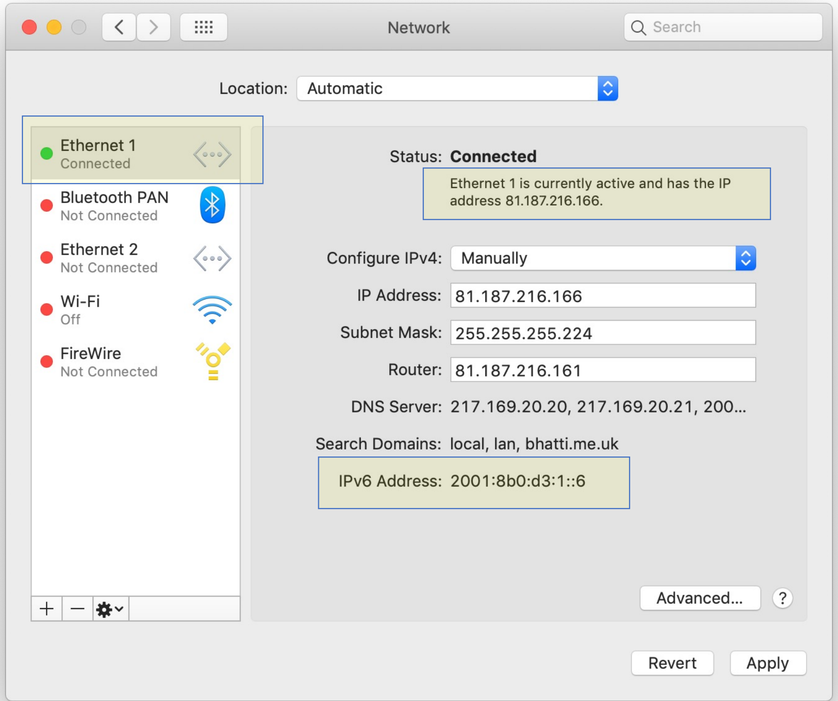
# A fundamental architectural constraint for IP

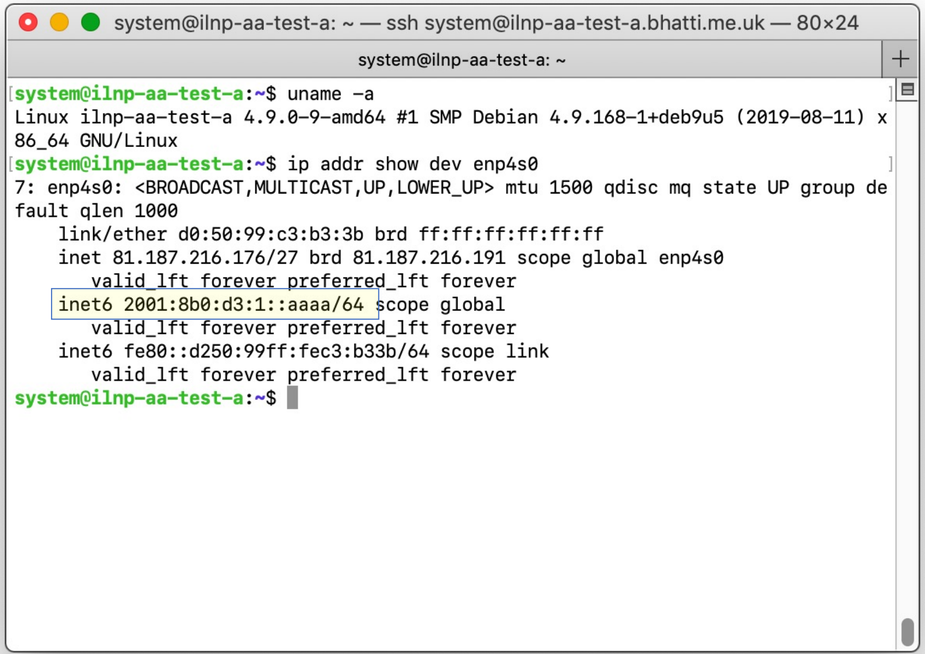
# Layered architecture and protocol "stack"



University of St Andrews

# Interfaces and addresses



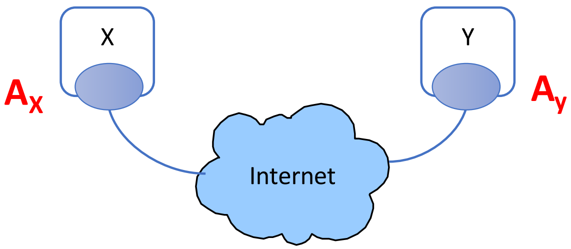



# End-to-end protocol state – IP addresses

saleem — -bash — 80×24

```
(base) falkland:~ saleem$ netstat -n -p tcp
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  81.187.216.166.22      162.244.77.140.41344   ESTABLISHED
tcp6       0      0  fe80::cd3:8a5:c9.50055 fe80::811:c2ae:d.50354 ESTABLISHED
tcp4       0      0  81.187.216.166.50020   52.111.236.11.443      ESTABLISHED
tcp4       0      0  81.187.216.166.22      191.223.53.11.34902    FIN_WAIT_1
tcp4       0      0  81.187.216.166.49984   54.239.32.228.443      ESTABLISHED
tcp6       0      0  2001:8b0:d3:1::6.49978 2606:4700:10::68.443   ESTABLISHED
tcp4       0      0  81.187.216.166.49972   99.86.114.24.443       ESTABLISHED
tcp6       0      0  2001:8b0:d3:1::6.49949 2603:1026:c06:23.443   ESTABLISHED
tcp6       0      0  2001:8b0:d3:1::6.49914 2a02:26f0:8f::17.443   ESTABLISHED
tcp4      31      0  81.187.216.166.49888   13.224.230.70.443      CLOSE_WAIT
tcp4       0      0  81.187.216.166.49879   52.58.102.8.443        ESTABLISHED
tcp4       0      0  81.187.216.166.49878   52.58.102.8.443        ESTABLISHED
tcp4       0      0  81.187.216.166.49872   104.75.173.25.443      ESTABLISHED
tcp4       0      0  81.187.216.166.49848   23.64.21.104.443       ESTABLISHED
tcp4       0      0  81.187.216.166.49824   88.221.176.116.443     ESTABLISHED
tcp4       0      0  81.187.216.166.49807   151.101.18.133.443     ESTABLISHED
tcp4       0      0  81.187.216.166.49780   23.64.43.119.443       ESTABLISHED
tcp4       0      0  81.187.216.166.49778   2.19.61.38.443         ESTABLISHED
tcp4       0      0  81.187.216.166.49777   151.101.18.133.443     ESTABLISHED
tcp4       0      0  81.187.216.166.49776   104.18.13.5.443        ESTABLISHED
tcp4      31      0  81.187.216.166.49773   99.86.116.88.443       CLOSE_WAIT
```

# Fundamental problem still remains …




A = IP address
P = port number

At X:
<TCP: $A_X$, $P_X$, $A_Y$, $P_Y$> <IP: $A_X$, $A_Y$>

At Y:
<TCP: $A_Y$, $P_Y$, $A_X$, $P_X$> <IP: $A_Y$, $A_X$>

| Protocol Layer | IP |
|---|---|
| Application | FQDN or IP address |
| Transport | IP address (+ port number) |
| Network | IP address |
| (Interface) | IP address |

**Entanglement ☹**

**Overloaded IP address semantics, e.g. transport layer communication is bound to a specific physical interface.**

FQDN fully qualified domain name

University of St Andrews

# "Ideal" address behaviour

**"IPv4 Address Behaviour Today".**
B. Carpenter, J. Crowcroft, Y. Rekhter.
RFC2101(I), Feb 1997.
https://datatracker.ietf.org/doc/html/rfc2101

```
3. Ideal properties.

   Whatever the constraints mentioned above, it is easy to see the ideal
   properties of identifiers and locators. Identifiers should be
   assigned at birth, never change, and never be re-used. Locators
   should describe the host's position in the network's topology, and
   should change whenever the topology changes.

   Unfortunately neither of the these ideals are met by IPv4 addresses.
   The remainder of this document is intended as a snapshot of the
   current real situation.
```

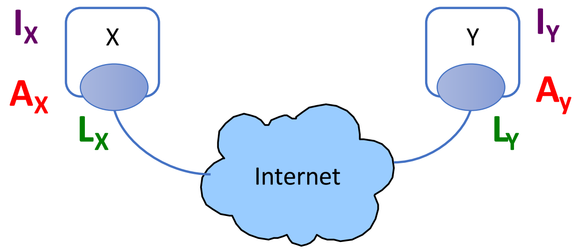
University of St Andrews

# Identifier-Locator Network Protocol (ILNP): cleaner naming and addressing for IP

# ILNP naming: identifiers and locators (1)

| Protocol Layer | IP | ILNP |
|---|---|---|
| Application | FQDN or<br>IP address | FQDN<br>(RFC1958) |
| Transport | IP address<br>(+ port number) | **(Node) Identifier**<br>(+ port number) |
| Network | IP address | **Locator** |
| (Interface) | IP address | (dynamic mapping) |
| | **Entanglement** ☹ | **Separation** ☺ |

FQDN fully qualified domain name




A = IP address
P = port number

At X:
<TCP: $A_X$, $P_X$, $A_Y$, $P_Y$> <IP: $A_X$, $A_Y$>

At Y:
<TCP: $A_Y$, $P_Y$, $A_X$, $P_X$> <IP: $A_Y$, $A_X$>

L = **Locator**
I = (Node) **Identifier**
**(I-LV = identifier-locator vector)**
P = port number

At X:
<TCP: $I_X$, $P_X$, $I_Y$, $P_Y$> <IP: $L_X$, $L_Y$>

At Y:
<TCP: $I_Y$, $P_Y$, $I_X$, $P_X$> <IP: $L_Y$, $L_X$>

University of St Andrews

# ILNP naming: identifiers and locators (2)

- **Locator, 64 bits, L64:**
  - **Is topologically significant.**
  - Names a (sub)network.
    same as today's **network prefix – good for routing.**
  - **L64 used only for routing and forwarding (network layer).**
- **Node Identifier, 64 bits, NID:**
  - **Is not topologically significant.**
  - Names a logical/virtual/physical node.
    does **not** name (bind to) an interface (dynamic binding).
  - **NID used only by upper layer protocols (e.g. transport layer).**

University of St Andrews

# Encoding L64 and NID values into IPv6: identifier-locator vector (I-LV)

**IPv6 address (as in RFC3587(I) + RFC4291(DS)):**

```
| 3 |     45 bits          | 8/16 bits |        64 bits              |
+---+----------------------+-----------+-----------------------------+
|       Unicast Routing Prefix         |    Interface Identifier     |
+---+----------------------+-----------+-----------------------------+
```

IPv6 routing (address) prefix

same syntax, different semantics

**ILNPv6 I-LV (as in RFC6741(E)):**

```
|              64 bits                 |        64 bits              |
+---+----------------------+-----------+-----------------------------+
|              Locator                 |   Node Identifier (NID)     |
+---+----------------------+-----------+-----------------------------+
```

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

# Packet view (network “wire image”)

## View from an IPv6 router

```
 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 IPv6 Address                       +
|                                                               |
+-                                                             -+
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                  Destination IPv6 Address                     +
|                                                               |
+-                                                             -+
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

## View from an IPv6 end-system

```
 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                      +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

University of St Andrews

# End-system OS kernel updates

- Updates required to end-system OS:
  - IPv6, ICMPv6 (control protocol), packet-handling paths, I-L bindings.
  - Transport level packet handling paths and PCB.
  - getaddrbyname(3) and related code (libc).
- **Existing socket(2) API works for well-behaved IPv6 applications:**
  - **IPv6 binaries can be used directly (see later).**
- Future – API that knows about ILNP:
  - benefits of using L64 and NID values directly.
  - ILNP could be "hidden" in higher layer frameworks/libraries, as socket(2) is today in many cases.

University of St Andrews

# ILNP follows “end-to-end” philosophy [SRC1984]

- **No NATs needed.**
- **No tunnels needed.**
- **No proxies / middleboxes needed.**
- **No changes to routing needed.**
- Harmonised functionality in the **end-system**, e.g.:
  - mobility without agents or proxies.
  - mobility and multihoming together (duality).
  - multihoming without extra routing state.
  - improvements to end-to-end packet-level security and privacy.
  - support for wide-area VM-image mobility.

*[SRC1984] J. H. Saltzer, D. P. Reed, D. D. Clark. 1984. End-to-end arguments in system design. ACM Trans. Comp. Sys, 2(4), Nov 1984, pp 277-288. https://doi.org/10.1145/357401.357402*

# Some experiments

# ILNP desktop testbed

- Emulate “real” network, “real” equipment. (mobility and multihoming/multipath).
- Only CN and MN ran new ILNP codebase:
  - extensions to Linux kernel v4.9 (LTS).
- All routers (R*x*) were IPv6 only:
  - backwards compatibility.
  - incremental deployment.
- MN physical interfaces (ethernet) turned on/off to emulate movement across the networks.
- Basic operation: TCP flows in progress between CN and MN during movement / multihoming activity.

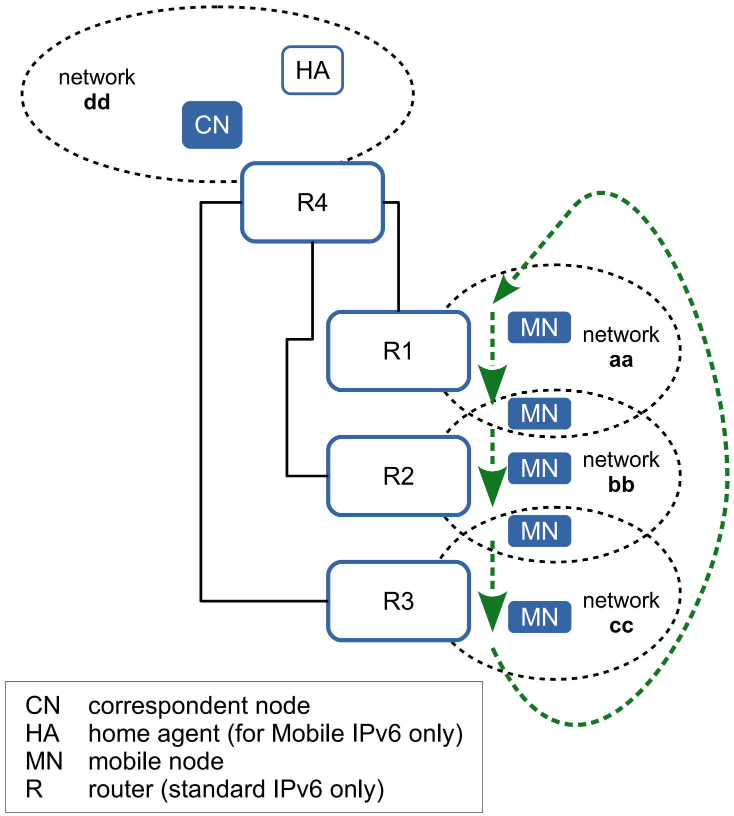
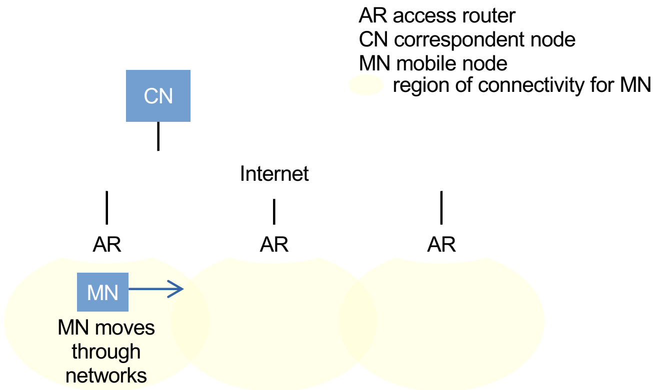



**Figure 6: The ILNP testbed for mobility experiments.**

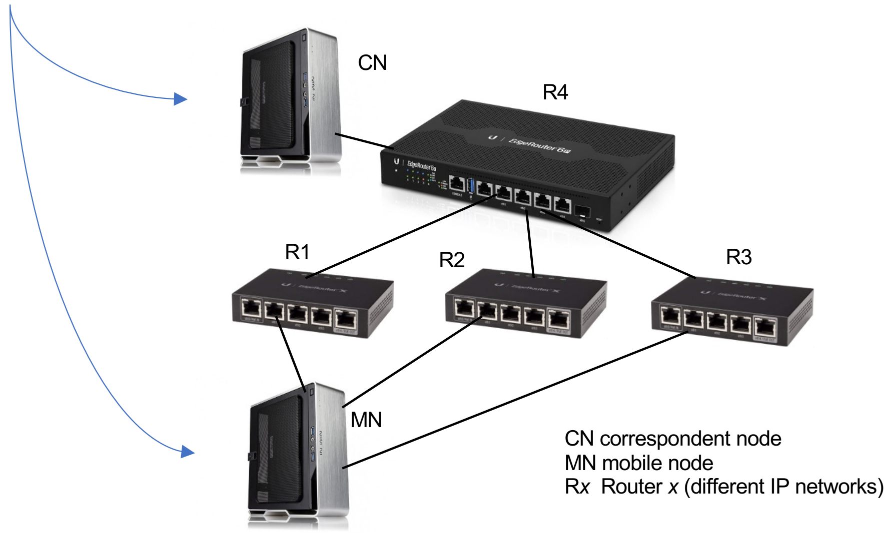
[YB2019] R. Yanagida, S. N. Bhatti. Seamless Internet connectivity for ubiquitous communication. PURBA2019, Pervasive Urban Applications Workshop (UBICOMP 2019 Conference). London, UK. 09 Sep 2019. https://doi.org/10.1145/3341162.3349315

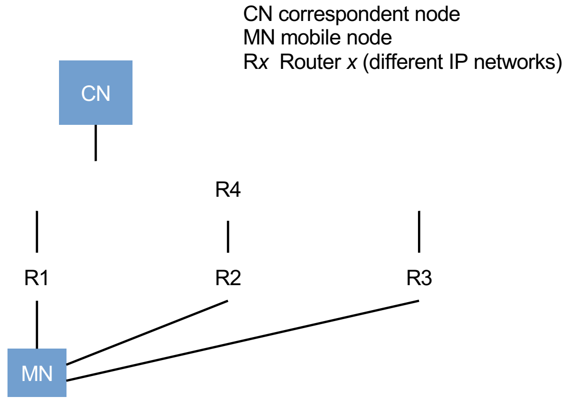
University of St Andrews

# Scenario and physical configuration




CN and MN running Modified Linux kernel



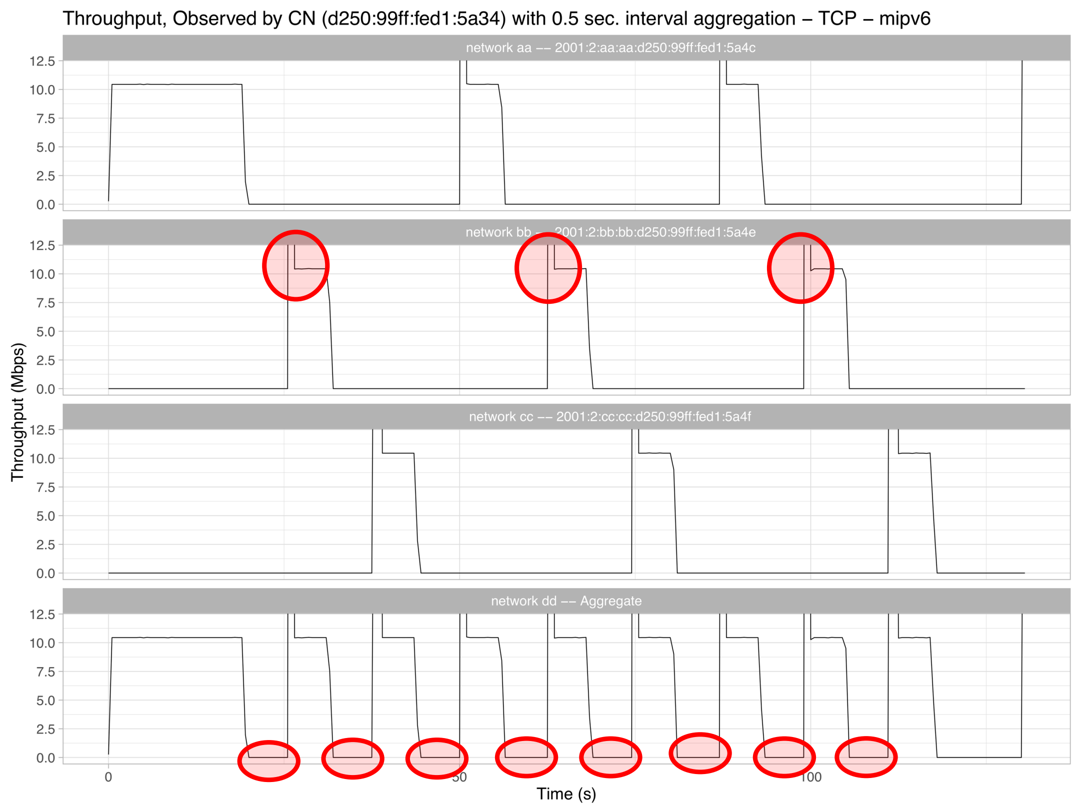
University of St Andrews

# Mobility experiment [YB2019]

[YB2019] R. Yanagida, S. N. Bhatti. Seamless Internet connectivity for ubiquitous communication. PURBA2019, Pervasive Urban Applications Workshop (UBICOMP 2019 Conference). London, UK. 09 Sep 2019. https://doi.org/10.1145/3341162.3349315

# Results – MIPv6

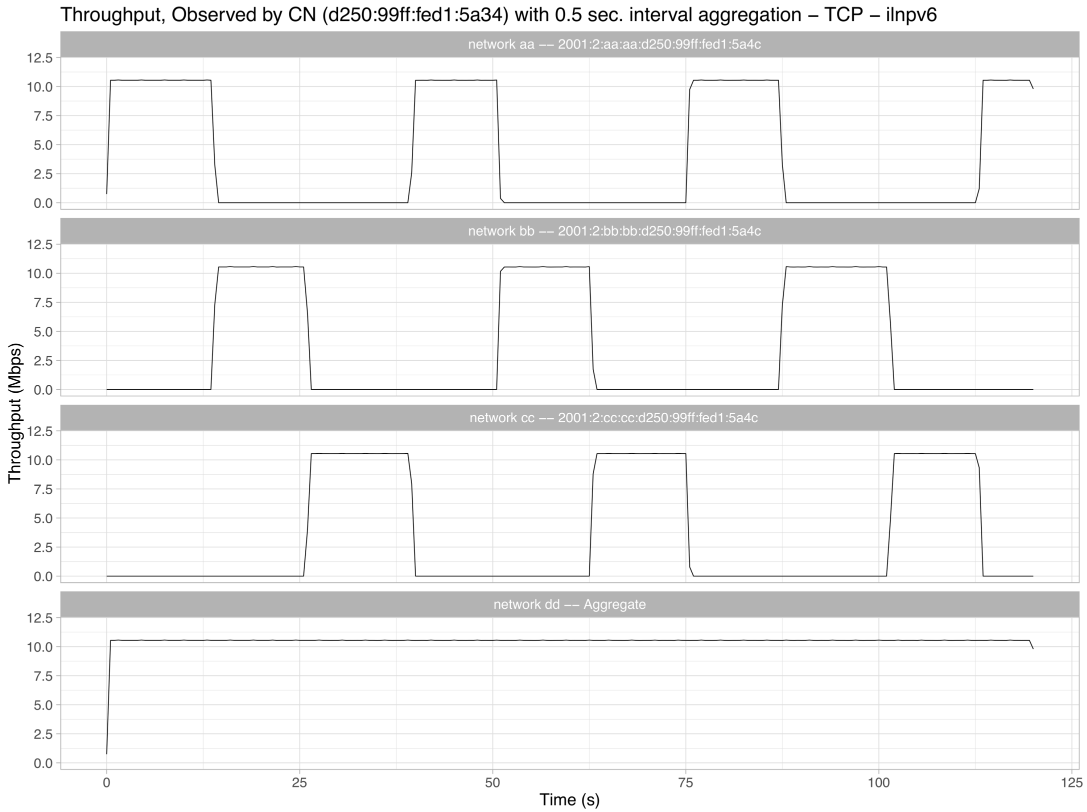
- (IETF recommended Internet solution for mobile nodes.)
- Explicitly overloads IP address semantics:
  - uses 2 addresses
  - HoA address (at HA)
  - CoA address (at MN)
- Loss of end-to-end transparency.
- Usual problems of proxy:
  - **performance**
  - security
  - privacy
  - scalability



University of St Andrews

# Results – ILNP

- Re-uses IPv6 packet format:
  - compatibility.
  - ease of deployment.
- Maintains end-to-end transparency.
- No issues of:
  - **performance**
  - (**scalability**)
- Also improves:
  - **security**.
  - **privacy**.



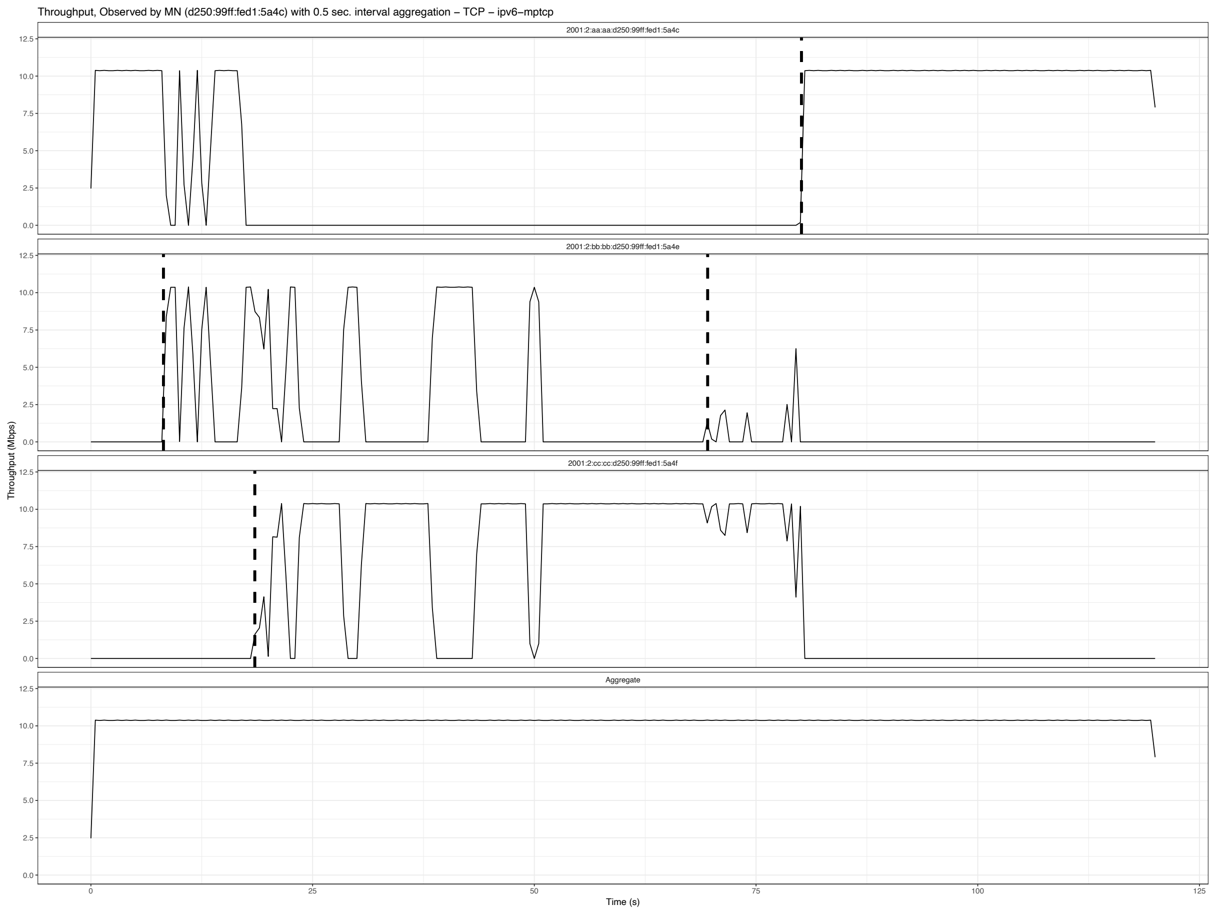
University of St Andrews

# Multipath experiment

Not yet published / peer-reviewed

# Results – Multipath TCP (MP-TCP) [RFC8684]

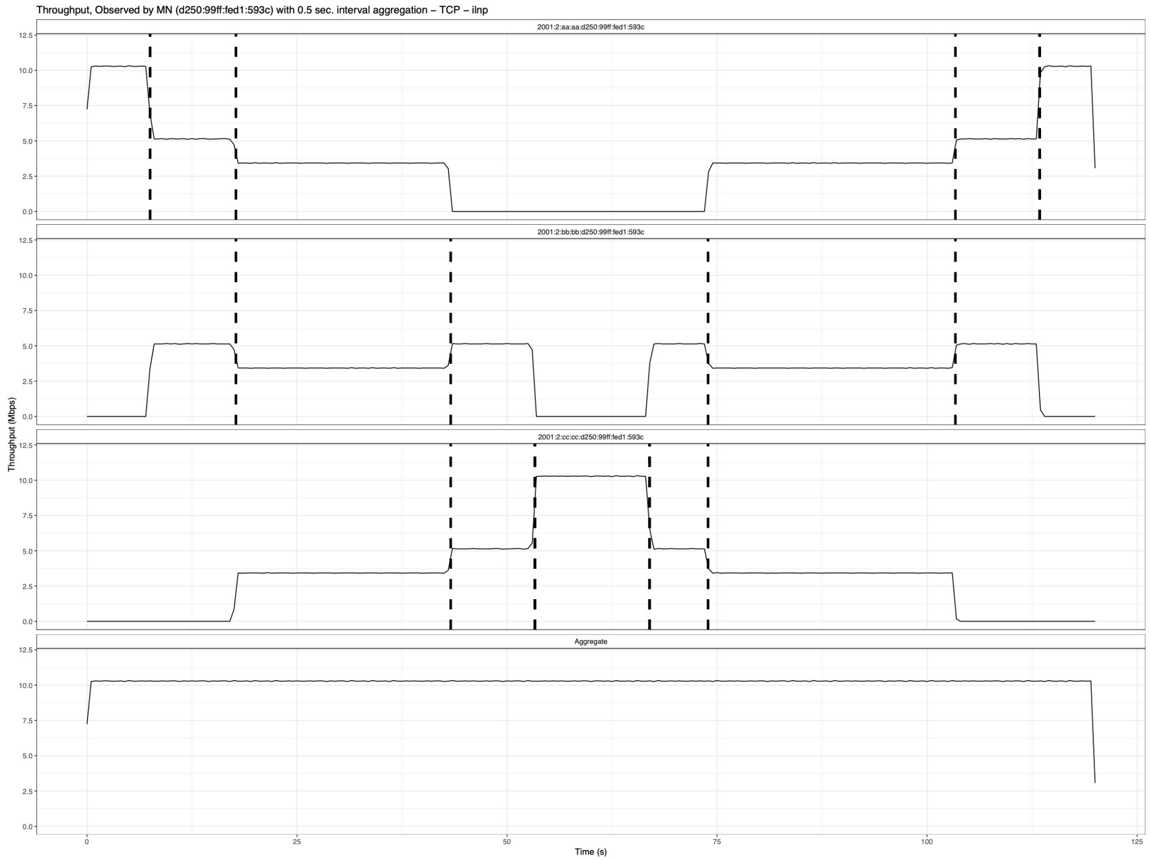
- Multipath TCP uses multiple addresses prefixes simultaneously.
- It is assumed each address represents and different path.
- Performs congestion control on each path.
- Packet distribution handled at transport layer (TCP).




[RFC8684] A. Ford, C. Raicui, M. Handley, O. Bonaventure, C. Paasch. "TCP Extensions for Multipath Operation with Multiple Addresses", RFC8684(PS), Mar 2020. https://datatracker.ietf.org/doc/html/rfc8684.html

# Results – ILNP with Linux default TCP

- Default TCP code in Linux kernel v4.9 (CUBIC).
- TCP is not "aware" of multipath, but has been modified to used multiple L64 values simultaneously.
- Packet distribution handled at network layer.



University of St Andrews

# Back to the question

# Are we stuck with IP?

- ILNP addresses a **long-standing problem** with the IP addresses:
  - A "refurbished hat" rather than "old hat"?
- The ILNP changes can be **deployed incrementally**:
  - Only end-systems that need to use ILNP need to be updated.
  - Looks like IPv6 on the wire for existing network equipment.
  - Can work with existing binaries without re-engineering or recompilation.
  - End-system updates could be pushed "over the air" as for OS updates today.
- **Opens up the revisiting of some other topics in networking research?**
- Not discussed in this talk:
  - DNS updates available in a number of existing DNS software, including BIND.
  - ICMPv6 and IPv6 header extensions look deployable (not published yet).

University of St Andrews

# Thank you!

- Saleem Bhatti: saleem@st-andrews.ac.uk
- ILNP: https://ilnp.cs.st-andrews.ac.uk/
- In progress / future work:
  - Real-time / video experiments.
  - Privacy and security.
  - New/extended socket(2) API, and “ILNP-aware” applications.
  - Extended use of DNS for applications.
  - FreeBSD 14-CURRENT code base (~Q1/2023).
  - (Plus others …)