Networking: Other Transports, NAT

i.g.batten@bham.ac.uk

Transports in wide use:

UDP是IP的一个封装，不可靠，也没有排序

- UDP: thin wrapper over IP, unreliable, unsequenced

TCP:完整的传输服务，提供可靠的、有序的交付，保证成功或积极的故障指示。

- TCP: complete transport service, offers reliable, sequenced delivery with guarantee of either success or a positive failure indication.

- Together majority of Internet traffic

这两个是互联网流量的绝大部分

RTP

- Real-time Transport Protocol 实时传输端口

  用于在某些应用程序中传输语音(电话)和视频流。

- Used to transport voice (telephony) and video (streaming) in some applications.

- Doesn’t do anything you can’t do yourself with UDP. UDP做不了的RTP也做不了

Problems for voice and video 声音和视频传输时的问题

- Consistent timing 一致的时间
- Choice between dropping and catching up
  在落下和追赶中作出选择
- Trade off with buffering 用缓冲来交换

对于电话传输来说

For telephony…

通常的说法是任何超过35ms的延迟对于会话来说都是有问题的

- Usual claim is anything over 35ms latency is problematic for conversation (“toll quality”)

  toll quality 收费质量 = 可以理解为有延迟的通话质量

  - Figure has no experimental basis 该数字没有实验基础
    - Partly about echo cancellation, partly about difficulty in maintaining conversation
      一部分是关于回声消除，一部分是关于维持对话的困难
- 35ms is easy to achieve in traditional telephone networks (roughly 10k km speed of light) but is difficult to achieve reliably in IP based networks with slow/ congested local links.

  35ms对于传统电话来说是很简单的（10KKm的光速），但对于基于IP的网络来说，在本地链路缓慢/拥挤的情况下很难可靠地实现

Reality is more generous

慷慨的，大方的

- Latency over networks with complex compression (“codecs”) is higher, GSM for example.
  在复杂压缩网络(“编解码器”)上的延迟更高，例如GSM。
  - Although GSM has no “side tone”, which is why people shout in mobile phones.
    虽然GSM没有“侧音”，这就是为什么人们在手机里大喊大叫。
- Increasingly, people will tolerate GSM-quality voice (~3kbps) rather than “toll quality” voice (~56kbps).
  越来越多的人会忍受gsm质量的声音(~3kbps)，而不是“toll quality”(有延迟)的声音(~56kbps)。
- Counter example is difficulty people have with geo-stationary satellite communications (ie 1960s/70s phone calls to Australia), but there latency approaches 500ms with heavy echo cancellation.
  相反的例子是人们在地球静止卫星通信上遇到的困难(比如60年代/70年代打到澳大利亚的电话)，
  但是有接近500ms的延迟和严重的回音取消。

RTP

bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Version	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID					Extension header length	
128+32×CC	Extension header ...						

RTP

每个包都包含一个序列号，序列号可以用来发现间隔并重新排序包。

- Each packet contains a sequence number, which can be used to spot gaps and re-order packets.

但是每个包还包含一个时间戳(设置流时的分辨率)

- But each packet also contains a time-stamp (resolution decided when the stream is set up)
  - Say, 8KHz for voice, as voice is most commonly 8KHz sampling rate, 4KHz bandwidth
    比如，8khz表示话音，因为话音通常是8khz的采样率，4khz带宽。
  - Or frame-rate for video
    或者视频的帧速率

与TCP的差异

Difference with TCP

- No acknowledgements. 没有确认
- Receiver knows when packet was sent, and how many were sent. 接受者知道什么时候包会被发送，以及有多少会被发送
- Receiver can therefore discard packets in order to stay “current”, or can pause replay to wait for arrival of missing packets, or some other strategy.
因此，接收者可以丢弃包以保持“当前”状态，或者暂停重播以等待丢失的包的到来，
或者其他一些策略。
- Duplicates are detected.
重复包是会被检测到的

RTP Setup

- RTCP (“real time control protocol”) used to set up video replay and similar

- SIP (“session initiation protocol” used to set up Voice over IP telephony.

- Co-ordination of RTCP/SIP session with RTP stream is difficult for firewalls: in voice-land, “Session Border Controllers” combine SIP and firewalling, while emptying your wallet.

- Most video streaming now uses traditional TCP with sufficient buffering to deal with variation in latency, plus heavy compression with MPEG/etc.

SCTP

- Stream Control Transport Protocol
- Attempt to tunnel traditional voice signalling (“SS7”) over internet.
- Again, UDP with a few extra facilities
- Largely moribund

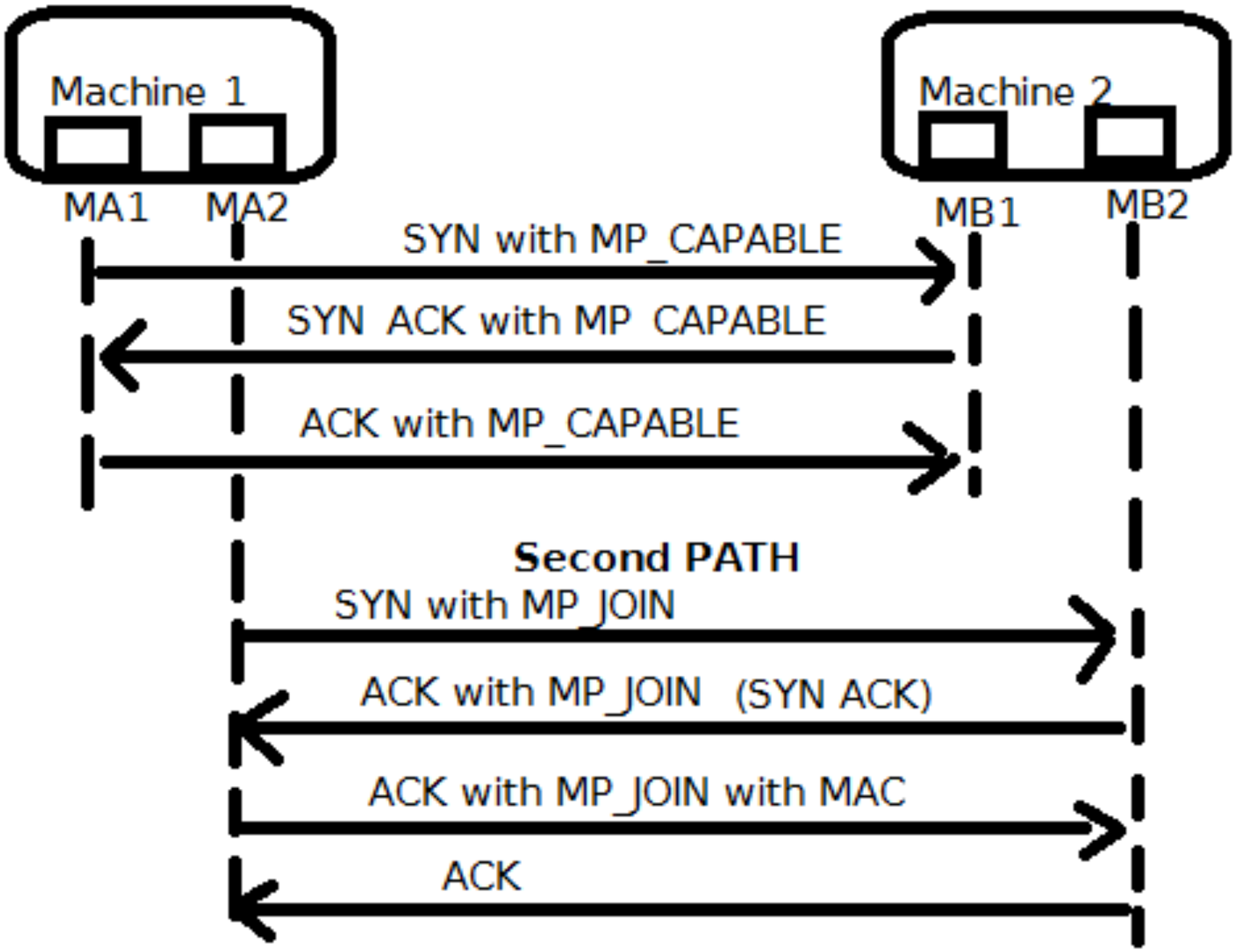
DCCP

- Datagram Congestion Control Protocol
- Another UDP plus frills, again for time-sensitive delivery.
- Again, moribund
- General lesson: “UDP plus a bit” is too complicated if it is general, insufficiently attractive to implementors if it is too specific.

Multipath TCP

- Now something more exciting!
- RFC6824 is well worth reading
- Allows multiple paths to be used by one TCP connection
  - For example, Wifi **and** 4G **simultaneously**

Multipath TCP



Not only performance

- By having a link multiplexed over WiFi and 4G, failure of one path appears as just some packet loss, and the link rapidly reconfigures.
    - This is very hard otherwise, as you will have different IP numbers in each realm
- Also makes effective use of multiple network cards, particularly in networks with a lot of resilience / redundancy.

New, but growing

- Implemented in iOS 7 et seq
- Reference implementation in Linux (much of the data centre world)
- Coming soon in Solaris (rest of the data centre world)
- Doesn’t require significant application changes, most applications work unmodified (may require recompilation)
- Looks promising

Address Translation

- Mechanism to extend scarce IP numbers
- Incidentally provides some security, although this was not a design goal and should be treated with care
- Breaks “end to end principle”
- Causes some people (such as me) to start shouting uncontrollably

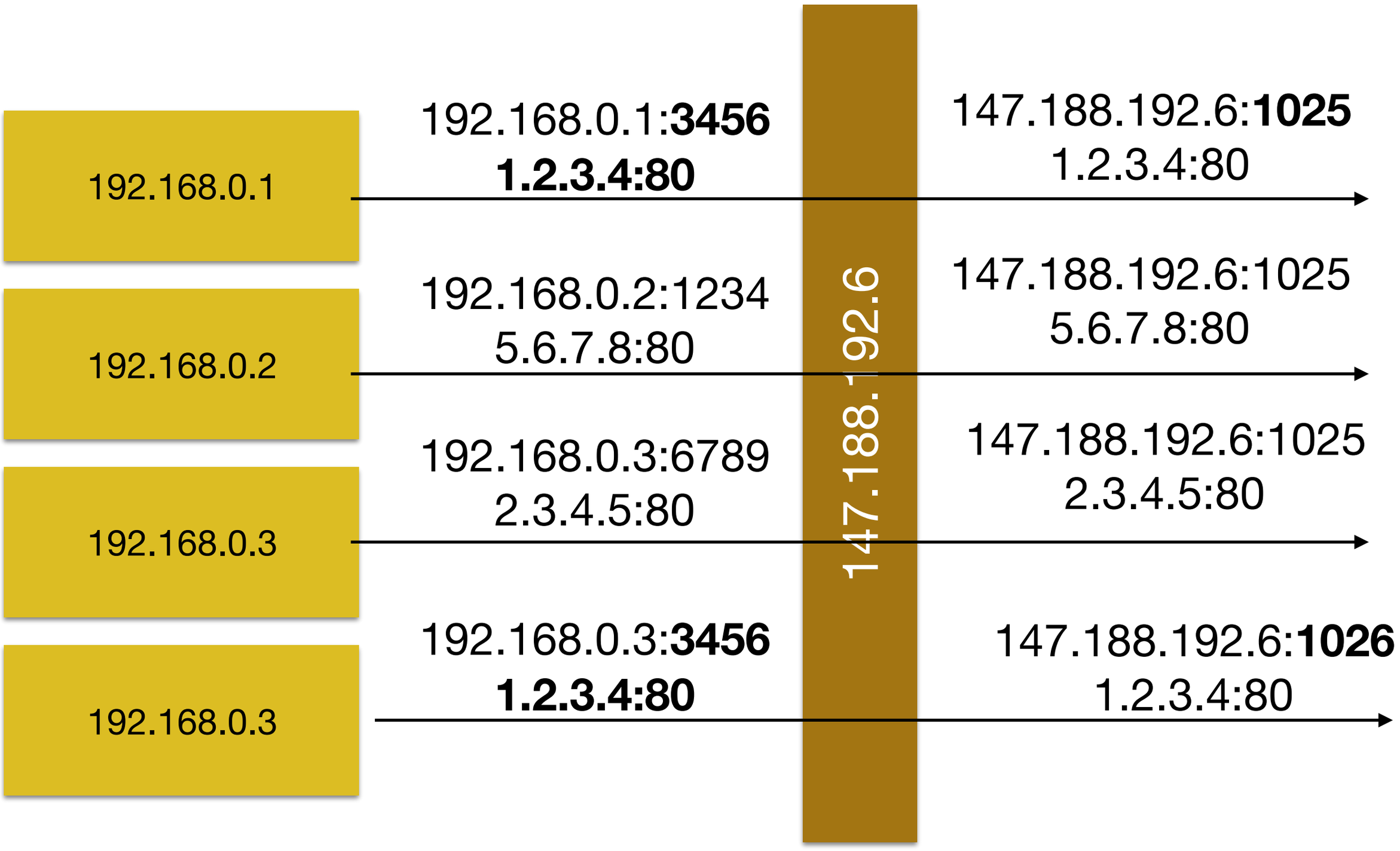
Basic Principles

- Outbound NAT:
  - Connection is modified so that connections from multiple source IP addresses are encoded into port number space of a smaller number of addresses
- Inbound NAT
  - Connection is modified so that connections to multiple ports on a small number of IP addresses are expanded out to a large number of addresses

Recall:

- TCP connection identified by source IP, source port, destination IP, destination port.
- So long as one element in the quad is different, it's a different (and distinguishable) connection
- Destination IP and port identify called service
- But the source can be changed

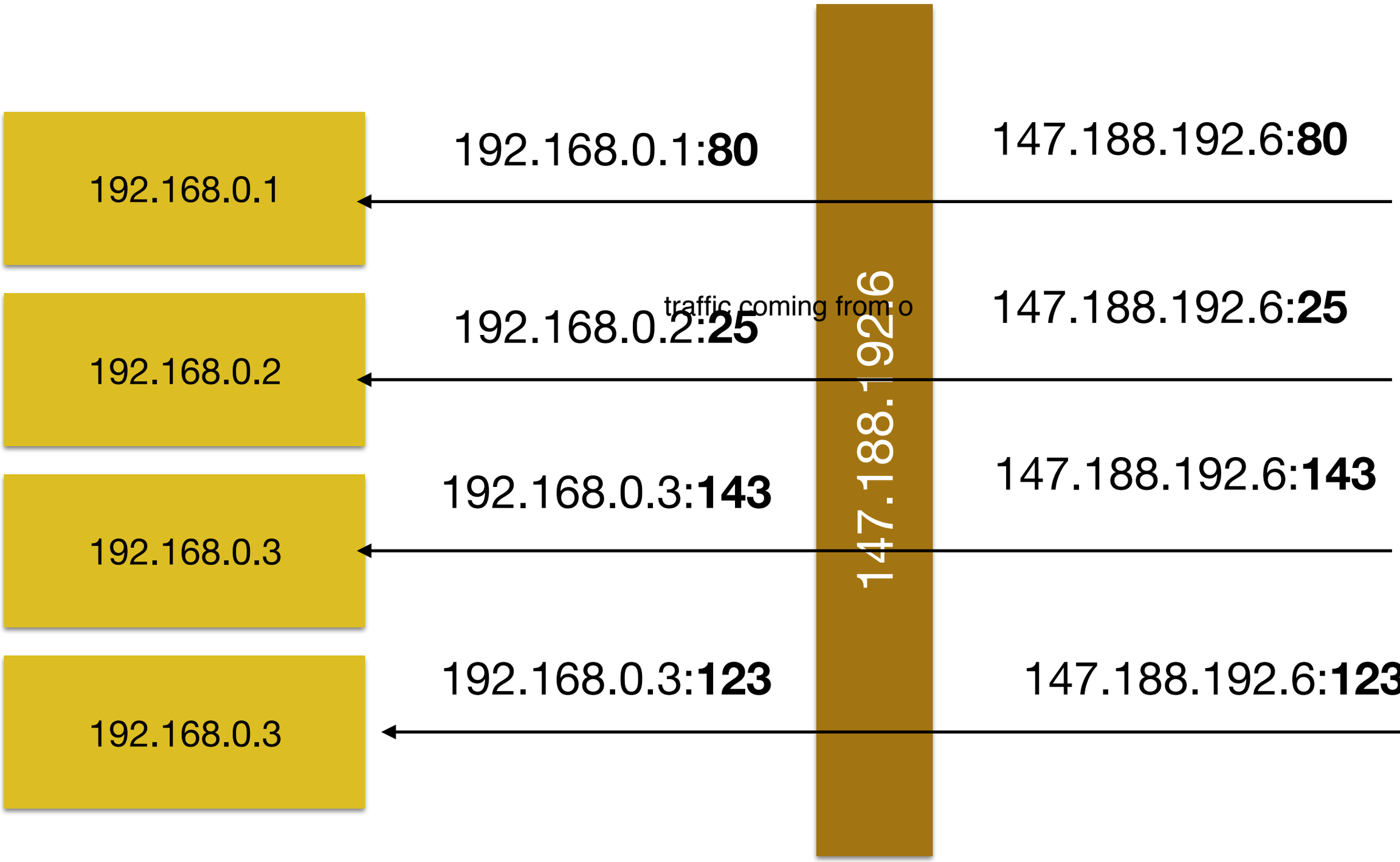
Outbound (Source) NAT



In reality…

- Often not necessary to overload port numbers as shown: each connection gets distinct source port number
  - Gives 65535 connections per IP number
- Large installations use multiple IP numbers at NAT point

Inbound (Destination) NAT



Inbound NAT

- Used to offer multiple services from single IP number (goes well with virtualisation to minimise attack surface)

- Also used in more complex situations to offer load balancing, failover, mobility, etc

NAT for TCP

- NAT device sees “SYN” packet and builds a mapping between inside and outside addresses.

- Modifies TCP packet (including IP header, as involves change to source address to be own), recomputes check sums, sends packet

change header again after recognising packet and sends to specific machine

- On receipt of packets, looks at source IP and port and destination port, performs reverse mapping and sends packet.

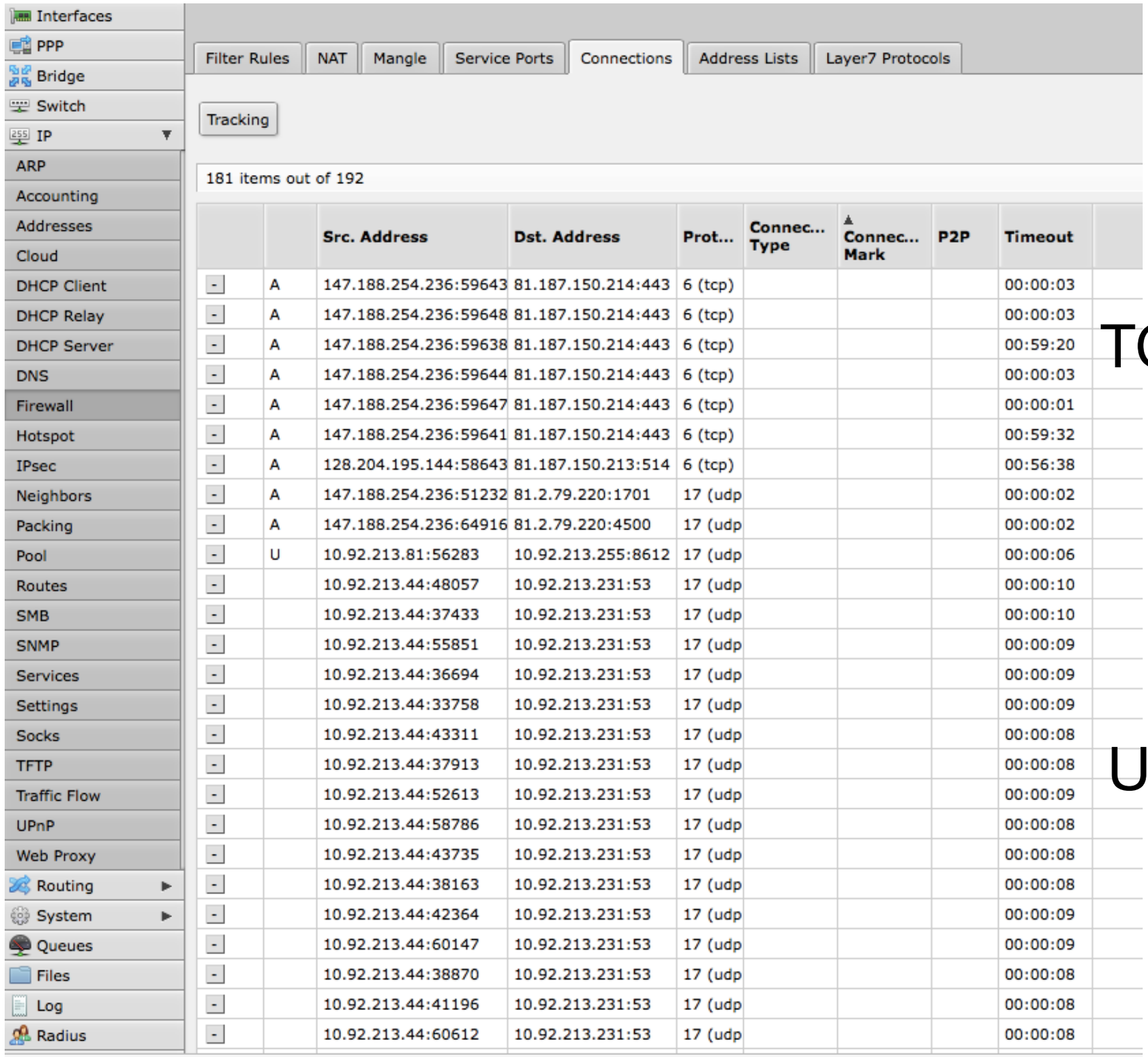
- Tracks TCP state, and deletes entry from translation table when FINs have all completed.

NAT for UDP

- No “state” as such.
- Rewrite outgoing UDP and then accept return packets until there is silence for 10s (typically).

UDP and NAT is tricky to choose

- Can also impose limit on number of replies, as for example DNS.



TCP, 60m timeout

UDP, 10s timeout

Problems with NAT

- It’s evil :-)
- Makes it very difficult to authenticate and log users

identify where traffic going. the source IP address are varies. login

log in

- NAT logging is part of “carrier grade NAT”, but requires time alignment of log on remote server and at the NAT point

Timing Problems

less of an issue now than it was

- my.popular.dom.ain server 1.2.3.4 has abusive connection from 147.188.192.6:1234 at 10:25:40

- 147.188.192.6 logging (if available) shows 1234 used for connections to 1.2.3.4 by 192.168.0.1 at 10:25:10 and 192.168.0.2 at 10:25:50.

  either requires that clocks are universally equivalent

- NAT logs won't include URL, just IP number

- Who called my.popular.dom.ain? Requires **retrospective** knowledge of clock offsets.

Logging Problems

- Most web logging does not record source ports. It can, but usually doesn't.

all you can tell by looking at log is some connection from any enterprise to me (server)

- So very difficult to request logs from NAT point, as there will be multiple connections to the same popular service, distinguished only by source port

- Claimed by law enforcement to be a serious problem.

Delays the IoT

- Internet of Things implies universal connectivity
- NAT delays universal connectivity, by making RFC1918 IP numbers usable for client devices.
- “Carrier Grade NAT” can even use RFC1918 for customer lines, NAT’d once at customer border and again at ISP border.

IPv6 has no NAT

- IPv6 does not require NAT, as plenty of addresses for everyone.

ipv6 can only get from ISP. people don't want to change v4
to v6 within an organization - huge process of changing ip numbers

- IPv6 implementations don't support NAT

- There are already proposals for IPv6 NAT, because of (bogus) security concerns.

NAT “Security”

- NAT is conceptually a stateful firewall: each TCP connection is being tracked for state, each UDP “connection” is being at least monitored for volume and duration

nearly all attacks done by trojan

- Tendency to regard this as an actual firewall, cf. PCI-DSS requirement for NAT on low-end companies.

credit card standard

- NAT products not certified or designed for security

- To complicate matters, often common code (Linux NAT functionality is in iptables firewall).

Inbound NAT

- This is particularly confusing for inbound NAT
- Inbound permits connection to port 80 on outside of NAT to appear as connection to port 80 on internal machine.
- There is **no security** in this at all: even if the NAT point is regarded as a firewall, this is a complete pass-through.
- Yet inbound NAT is still used as a “security” feature.

Complications for NAT

- Protocols which embed IP numbers in control streams break under NAT, because the IP numbers are wrong.

- FTP is the worst offender, and requires custom NAT modules to re-write the contents of the control stream.

- Modified FTP (“Passive Mode”, “PASV”) is better solution, or just don’t use FTP (please, just don’t use FTP).

Complications for NAT

- IP-address based authentication schemes lose resolution, because all of a site appears as one address.

cannot do any access control based on IP numbers as we cannot distinguish between machines

- Such schemes were arguably broken anyway, but are popular in academic publishing. Solutions involve complex proxying, but real solution is better authentication strategies.

Extra NAT protocols

- UPnP (“Universal Plug ’n’ Play” — who, one has to ask, names these protocols?)

- Allows “inside” devices to communicate with a NAT point and request inbound NAT, effectively automating a bypass of any firewall.

NAT getting completely out of control. First used by games boxes to NAT network for video game VPN without any authentication any device inside network can open

- Used heavily in residential products like Web Cams and “personal cloud” type products, as well as VoIP.

- UPnP is a dream for malware, as it makes opening a connection to a command and control server particularly easy.

Summary

- Quite a few alternatives to TCP and UDP, mostly used only for voice.
- Multipath TCP looks very promising.
- NAT is a necessary evil, but please, IPv6.