Implementation of NAT

A NAT router must (transparently) perform the following:

1. **For all outgoing datagrams**: replace (source IP address, port #) to (NAT IP address, new port #). Remote clients/servers will perceive (NAT IP address, new port #) as the end host they are communicating with, and will address their packets to that.

2. **Maintain a NAT translation table**: record all mappings from (source IP address, port #) to (NAT IP address, new port #) in a look up table.

3. **For all incoming datagrams**: replace (NAT IP address, new port #) in destination field of every incoming datagram with the corresponding (source IP address, port #) stored in NAT table.

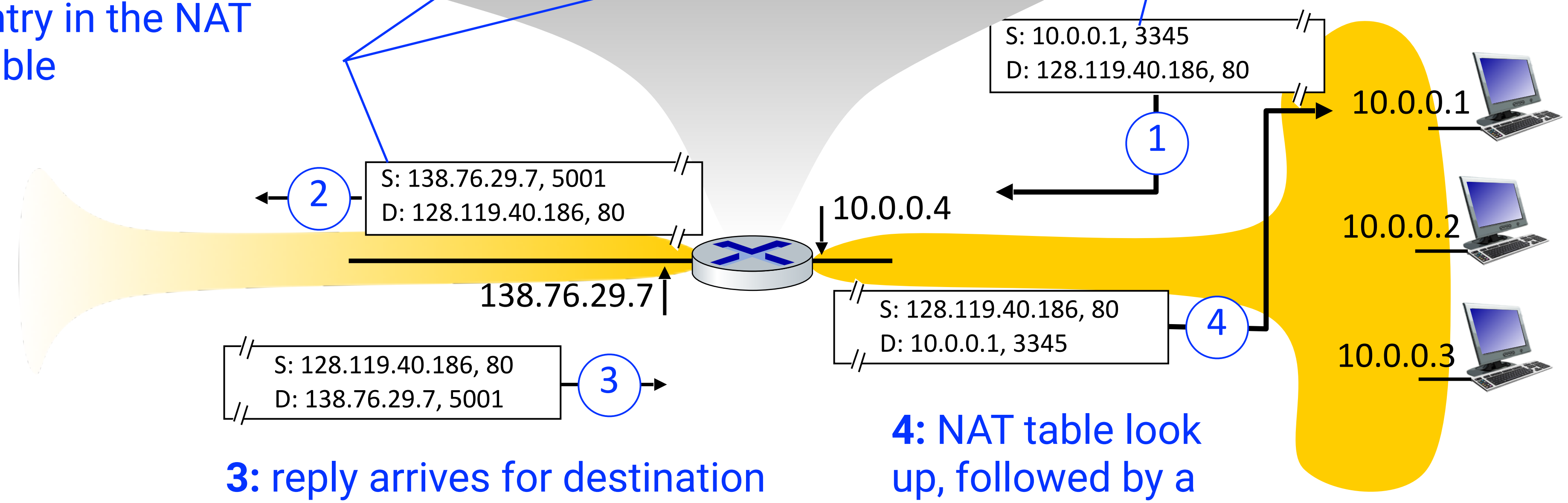
Implementation of NAT

2: NAT router changes datagram source address from 10.0.0.1 port 3345 to 138.76.29.7 port 5001; creates an entry in the NAT translation table

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

1: host 10.0.0.1 sends datagram to 128.119.40.186 port 80



3: reply arrives for destination address: 138.76.29.7 port 5001

4: NAT table look up, followed by a source IP and port number rewrite

Network Address Translation (NAT)

All devices in local network can have addresses from the "private" IP address space (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) that can only be used in local network

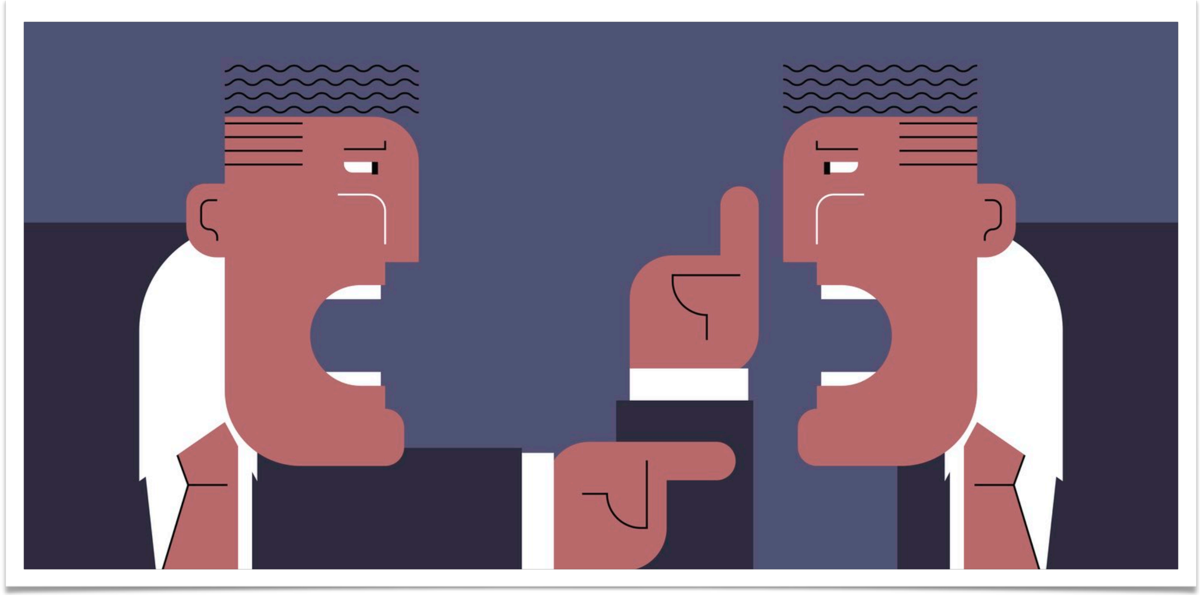
How is this useful?

- just one IP address needed from provider ISP for all devices
- can change addresses of host in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- [bonus] **security**: devices internal to the local network are neither directly addressable nor visible to the outside world

Since early days till now

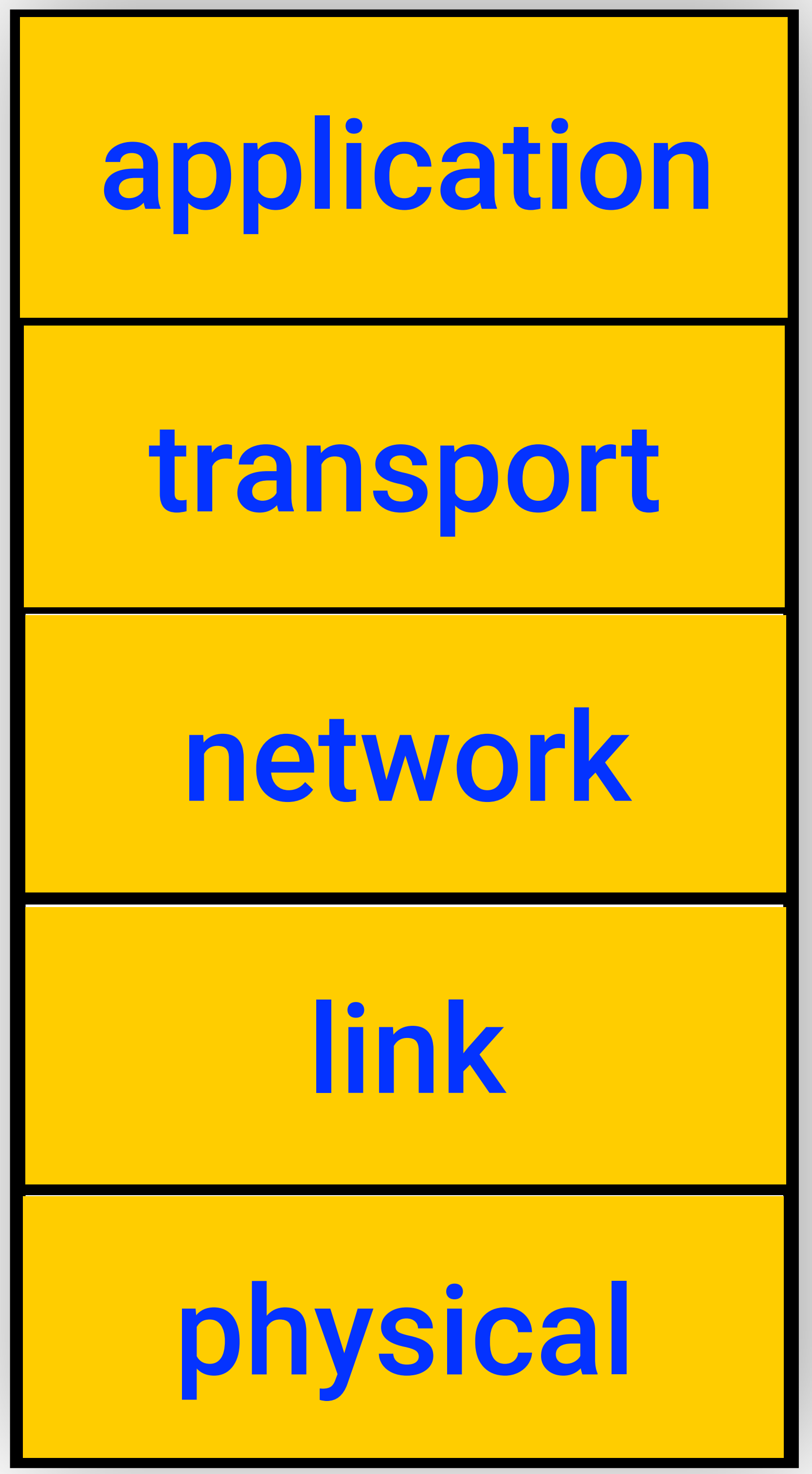
NAT has been

CONTROVERSIAL



- routers “should” only process up to layer 3
- address “shortage” should be solved by IPv6
- violates end-to-end argument (port # manipulation by network-layer device)
- NAT traversal: what if client wants to connect to a server that is behind NAT?

The five layer architecture of the Internet



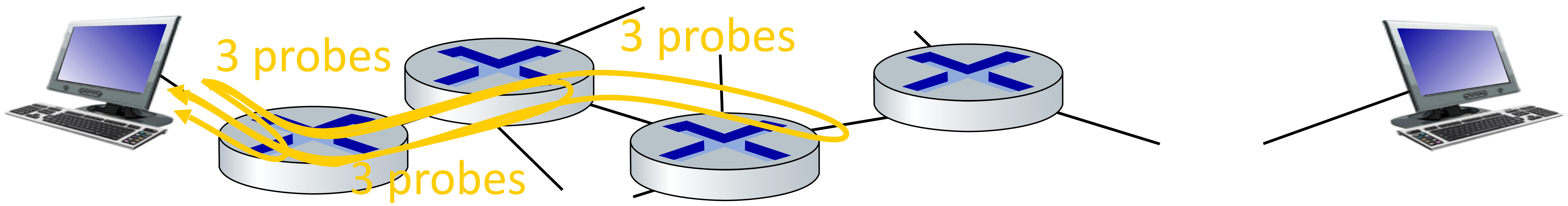
- **Application**: *supporting network applications. E.g., HTTP, IMAP, SMTP, DNS*
- **Transport:** *process to process data transfer. E.g.,TCP, UDP*
- **Network:** *routing of datagrams from source machine to destination. E.g., IP, IPv6*
- **Link:** *deliver data between neighboring network elements. E.g., Ethernet, 802.11 (WiFi)*
- **Physical:** *bits "on the wire". E.g., 10BASE-T*

Quantifying delays in the "real" Internet

traceroute: *a tool that provides delay measurement from source to router along end-end Internet path towards destination.*

For all i

(i) *send three packets that will reach router i on path towards destination (with time-to-live field set to i)*
(ii) *router i will return packets to sender*
(iii) *sender measures time interval between transmission and reply*



Quantifying delays in the “real” Internet

traceroute: from fastx01.divms.uiowa.edu to www.google.com

Router hierarchy within uiowa

uiowa’s ISP

3-delay measurements

```
[sshastri@fastx01 sshastri]$ traceroute www.google.com
traceroute to www.google.com (172.217.8.164), 30 hops max, 60 byte packets
 1  rtr-lc-1-production.net.uiowa.edu (128.255.58.1)  0.733 ms  0.776 ms  0.966 ms
 2  rtr-core-lc.net.uiowa.edu (128.255.2.44)  0.529 ms  0.626 ms  0.818 ms
 3  rtr-border-bsb.net.uiowa.edu (128.255.2.225)  0.377 ms  0.393 ms  0.406 ms
 4  r-equinix-isp-ae0-2236.wiscnet.net (216.56.50.73)  5.406 ms  5.406 ms  5.410 ms
 5  72.14.218.180 (72.14.218.180)  5.423 ms  5.426 ms  5.476 ms
 6  108.170.244.1 (108.170.244.1)  5.317 ms 108.170.243.225 (108.170.243.225)  6.289 ms  6.292 ms
 7  72.14.232.153 (72.14.232.153)  5.386 ms  5.346 ms  5.343 ms
 8  ord37s08-in-f4.1e100.net (172.217.8.164)  5.306 ms  5.296 ms  5.292 ms
[sshastri@fastx01 sshastri]$
```

Who is answering for google?

Why aren’t the delays strictly increasing?