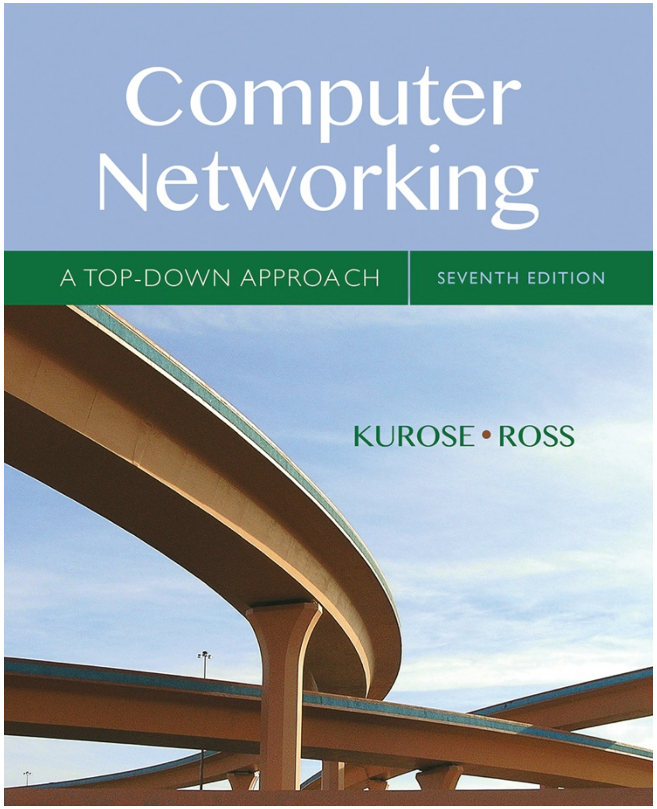
Chapter 4
Network Layer: The Data Plane



A note on the use of these Powerpoint slides:

We' re making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we' d like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach

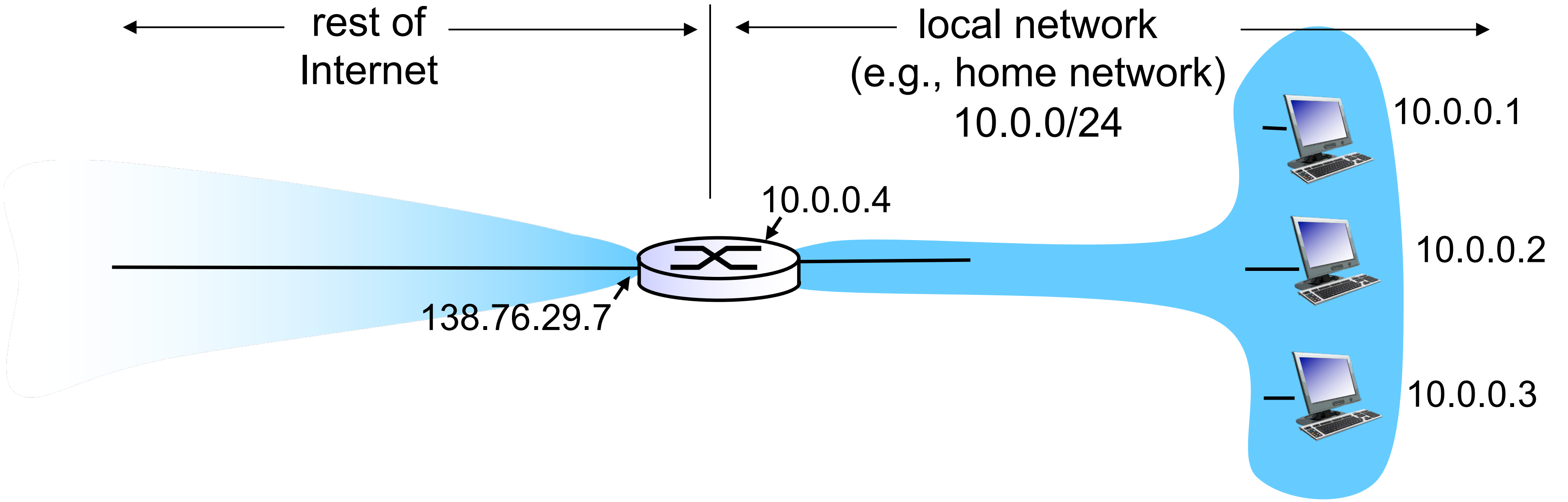
7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Minor modifications made to original slides by Nathan Bowman

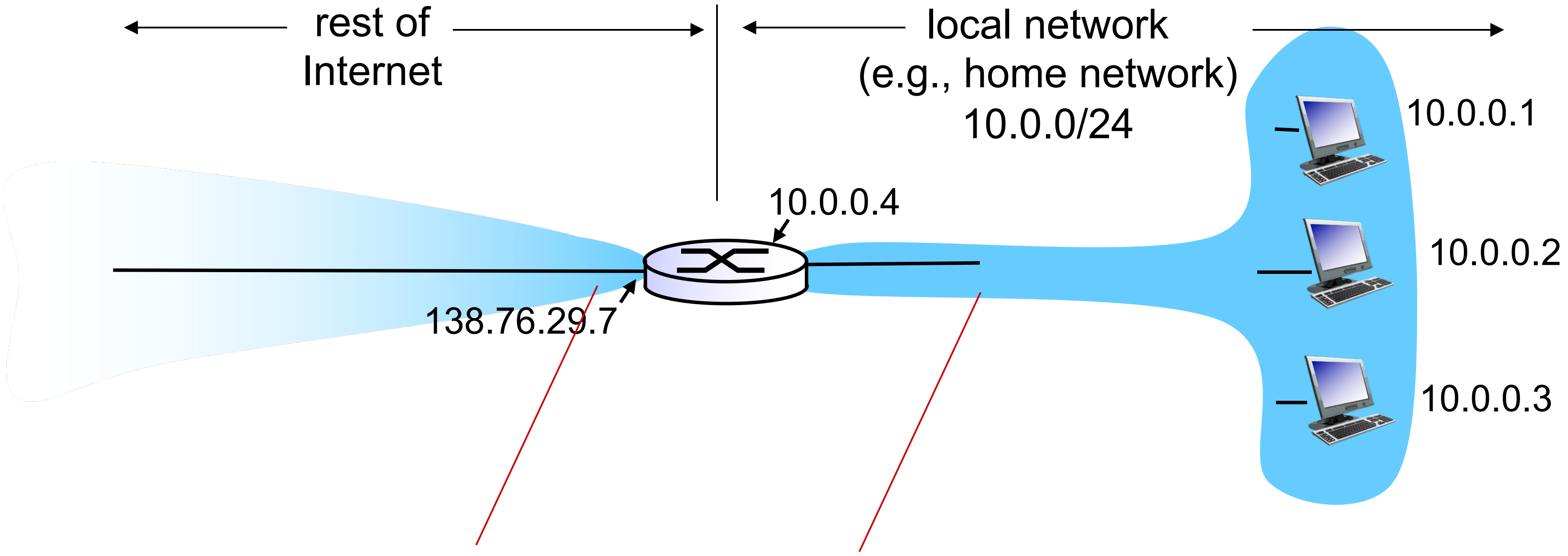
NAT: network address translation

- 2^{32} addresses is *a lot*
- But, the internet is very big
- New protocol, IPv6, has more addresses, but not yet widely adopted
- In the meantime one way to get more addresses (in addition to some other benefits) is to use NAT

NAT: network address translation



NAT: network address translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

motivation: local network uses just one IP address as far as outside world is concerned:

- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT: network address translation

implementation: NAT router must:

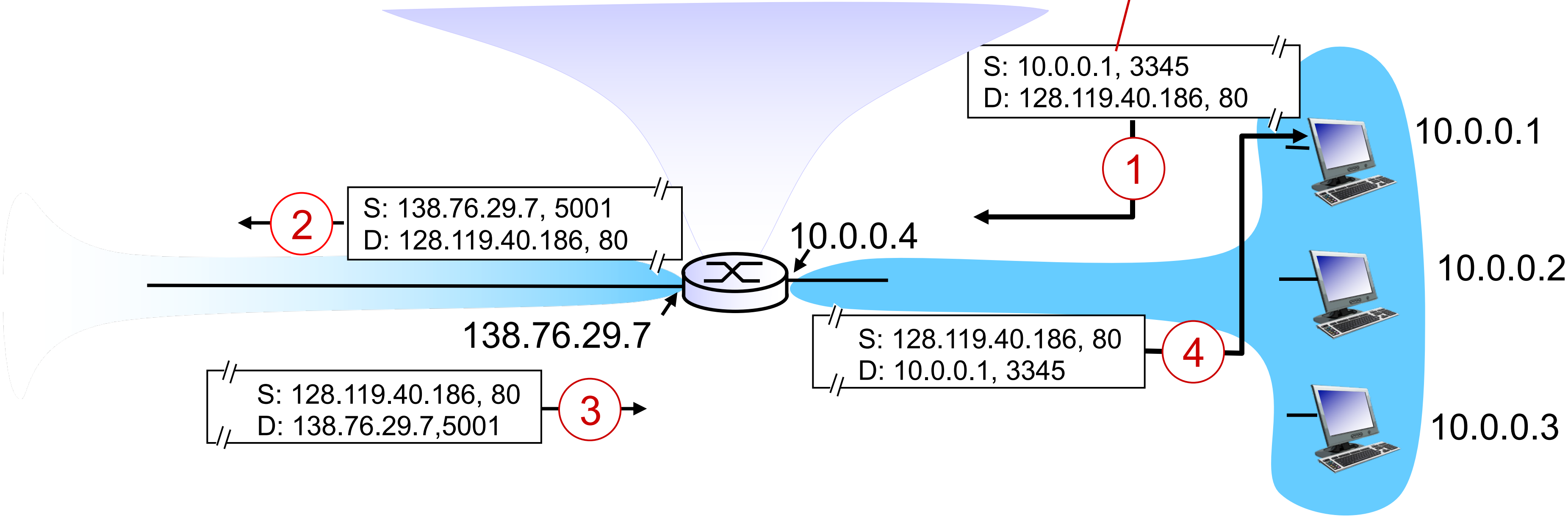
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

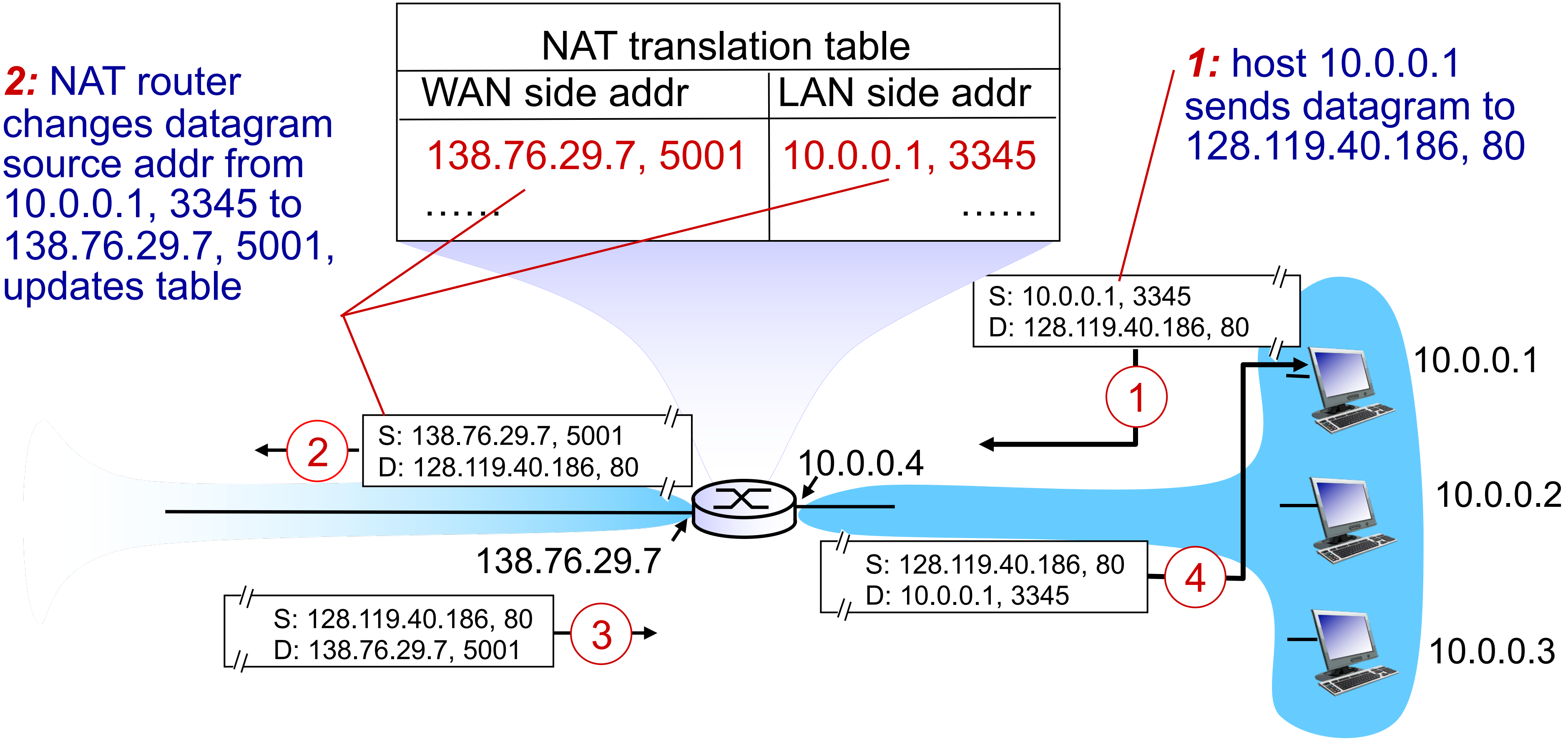
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation



1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

NAT: network address translation



2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

NAT translation table

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

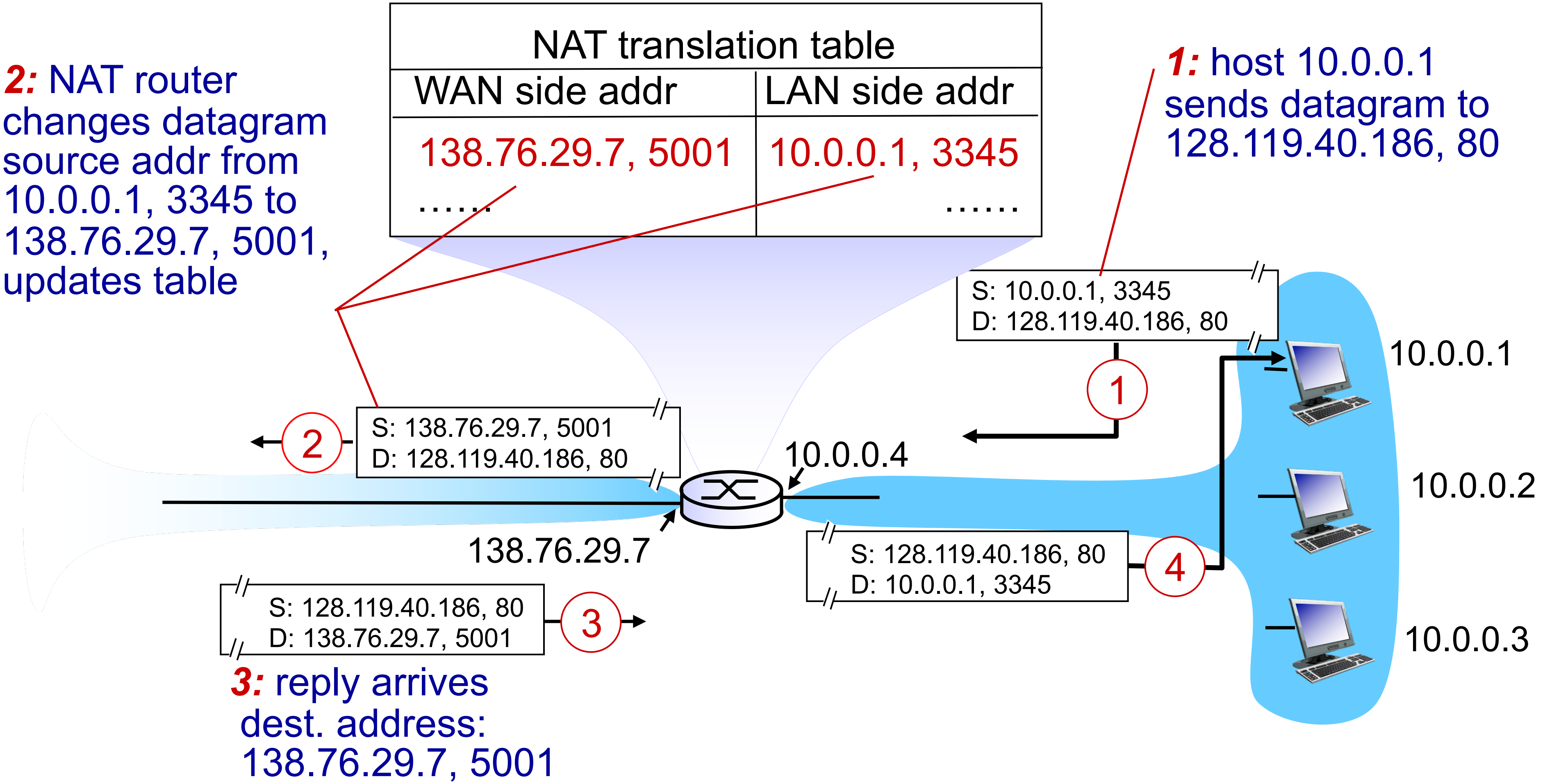
S: 10.0.0.1, 3345
D: 128.119.40.186, 80

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

NAT: network address translation



NAT translation table

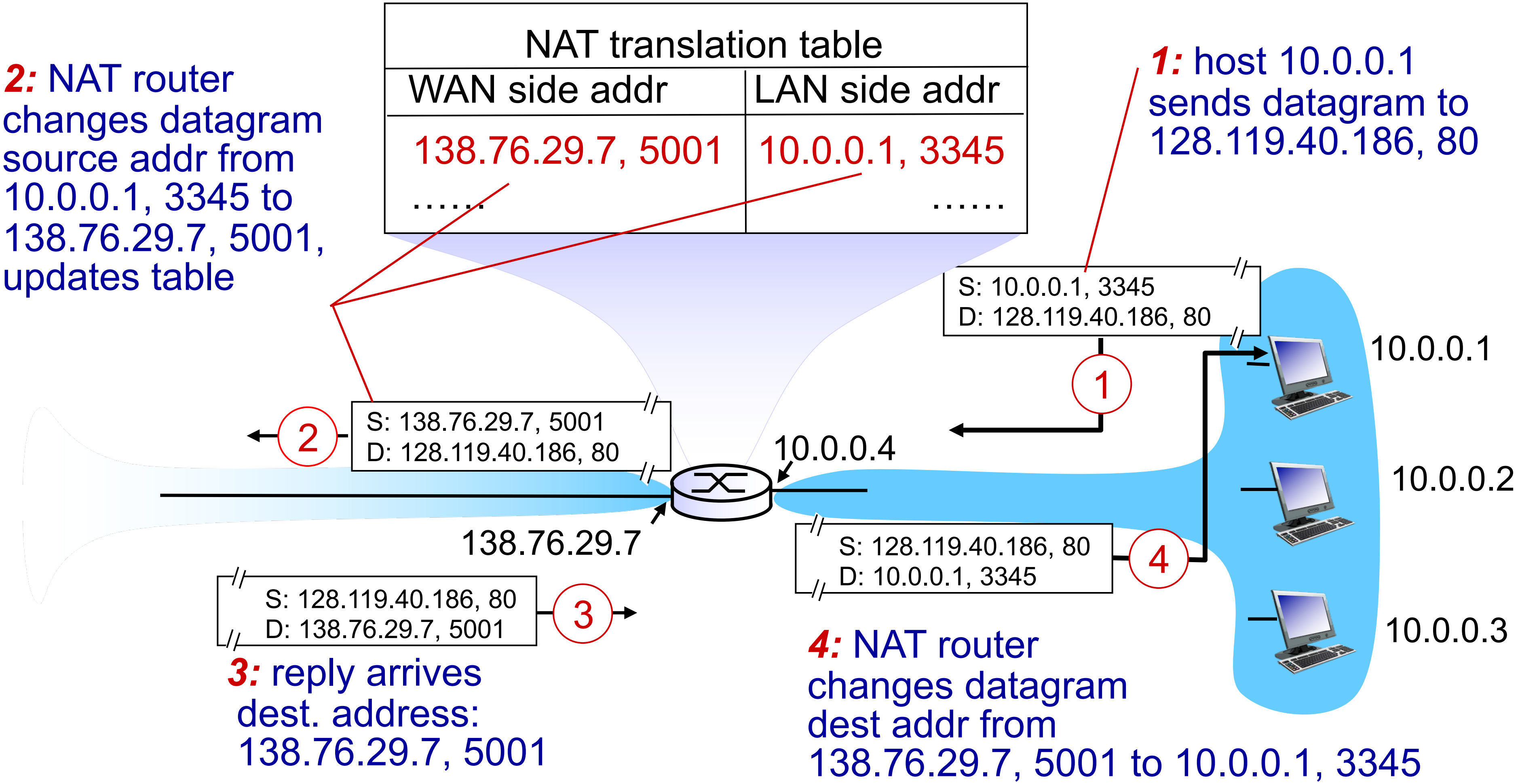
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

3: reply arrives dest. address: 138.76.29.7, 5001

NAT: network address translation



2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates 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, 80

S: 10.0.0.1, 3345
D: 128.119.40.186, 80

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

3: reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT: network address translation

- All of this is transparent to both local and remote hosts
- Hosts on subnet "believe" they are host on public internet with address such as 192.168.1.2
- Servers replying to messages from NAT "believe" they are communicating directly with host at e.g., 138.76.29.7
- Router at edge of NAT is only device that knows true situation

NAT: network address translation

- Addresses are not visible to outside world
- Some ranges of IP addresses reserved for use on private network
  - 10.0.0.0 to 10.255.255.255 — i.e., 10.0.0.0/8
  - 172.16.0.0 to 172.31.255.255 — i.e., 172.16.0.0/12
  - 192.168.0.0 to 192.168.255.255 – i.e., 192.168.0.0/16
- There can be no hosts on public internet with addresses in those range
- Due to NATs, many devices around the world use the same IP address at once (but only from perspective of local network)

NAT: network address translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - address shortage should be solved by IPv6
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - NAT traversal: what if client wants to connect to server behind NAT?

NAT: network address translation

- NAT as described cannot handle *incoming* connections
- When outside world sends message to 138.76.29.7, how would router know which local host is actually targeted?
- Solution is for hosts to register ahead of time and router to implement **port forwarding**
- With port forwarding, router keeps entries in its NAT translation table for specific port-to-local-host mappings

NAT: network address translation

- For example, assume host with local address 10.0.0.2 wishes to run web server behind NAT
- Router configured to forward all incoming traffic addressed to 138.76.29.7:80 to 10.0.0.2:80
- If another host wishes to run web server, needs to advertise on different port on router, but can still use port 80 locally
  - For example, forward 138.76.29.7:8080 to 10.0.0.3:80
- Also useful setting up remote ssh, game servers, etc.