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

All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers.
• **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).
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
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

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

3: Reply arrives dest. address: 138.76.29.7, 5001

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345
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 violates end-to-end argument

• NAT possibility must be taken into account by app designers, eg, P2P applications

address shortage should instead be solved by IPv6
client wants to connect to server with address 10.0.0.1
server address 10.0.0.1 local to LAN (client can’t use it as destination addr)
only one externally visible NATted address: 138.76.29.7

solution 1: statically configure NAT to forward incoming connection requests at given port to server
e.g., (123.76.29.7, port 2500)
always forwarded to 10.0.0.1 port 25000
solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:

- learn public IP address (138.76.29.7)
- add/remove port mappings (with lease times)

i.e., automate static NAT port map configuration
solution 3: relaying (used in Skype)
NATed client establishes connection to relay
External client connects to relay
relay bridges packets between to connections
IPv6

- **Initial motivation:** 32-bit address space soon to be completely allocated.

Additional motivation:
header format helps speed processing/forwarding
header changes to facilitate QoS

IPv6 datagram format:
fixed-length 40 byte header
no fragmentation allowed
**Priority:** identify priority among datagrams in flow

**Flow Label:** identify datagrams in same “flow.”
(concept of “flow” not well defined).

**Next header:** identify upper layer protocol for data
Other Changes from IPv4

- **Checksum**: removed entirely to reduce processing time at each hop
- **Options**: allowed, but outside of header, indicated by “Next Header” field
- **ICMPv6**: new version of ICMP additional message types, e.g. “Packet Too Big”
  multicast group management functions
Not all routers can be upgraded simultaneous
no “flag days”
How will the network operate with mixed IPv4 and IPv6 routers?
• **Tunneling**: IPv6 carried as payload in IPv4 datagram among IPv4 routers
Tunneling

Logical view:

Physical view:
Tunneling

Logical view:

Physical view:

Flow: X
Src: A
Dest: F
data

Flow: X
Src: A
Dest: F
data

Flow: X
Src: A
Dest: F
data

Flow: X
Src: A
Dest: F
data

A-to-B:
IPv6

B-to-C:
IPv6 inside
IPv4

E-to-F:
IPv6

B-to-C:
IPv6 inside
IPv4

IPv6
IPv6
IPv6
IPv6
IPv6
IPv6
IPv6
IPv6
IPv6
IPv4
IPv4
IPv4
IPv4
IPv6
IPv6
IPv6
IPv6
IPv6