Other approaches to solve address depletion problem:

• IPv6

→ 128-bit addresses

→ who wants it (or doesn't want it)?

→ IPv4 still dominant

IPv4 has found real-world workarounds limiting necessity of IPv6 deployment

→ repurposing of existing resources

→ IPv6: complexity and overhead

→ backward compatibility and cost

IPv6 header format:

<table>
<tr><td colspan="1">version
4</td><td colspan="2">traffic class
8</td><td colspan="5">flow label
20</td></tr>
<tr><td colspan="4">payload length
16</td><td colspan="2">next header
8</td><td colspan="2">hop limit
8</td></tr>
<tr><td colspan="8">source address
128</td></tr>
<tr><td colspan="8">destination address
128</td></tr>
</table>

- traffic class: similar role as TOS field in IPv4
- flow label: flow label + source address

  → per-flow traffic management

  → significant extra bits

  → header size twice as large: 40 bytes

- next header: similar to IPv4 protocol field

  → plus double duty for option headers

  → integrated with IPsec: authentication, encryption

- hop limit: same role TTL

- missing fields

  → fragmentation header optional: only allowed at source

Key features of IPv4 global Internet:

- Classless (vs. classful) IP addressing

→ variable length subnetting

→ that is, $a.b.c.d/x$ (x: mask length)

→ e.g., 128.10.0.0/16, 128.210.0.0/16, 204.52.32.0/20

Prefix specifies organization: autonomous system

→ IPv4 (and IPv6) addresses allocated to autonomous systems

→ Purdue University: ASN 17

→ AT&T: ASN 17

→ used in inter-domain routing

→ CIDR (classless inter-domain routing)

→ de facto global Internet addressing standard

- Dynamically assigned IP addresses

→ share an IP address pool

→ reusable

→ e.g., DHCP (dynamic host configuration protocol)

→ UDP-based client/server protocol (ports 67/68)

→ used in access ISPs, enterprises, home networks, etc.

→ customer premises equipment: almost persistent IPv4 addresses

Note: WLANs, cellular connections, modem dial-up connections, etc. are more dynamic, temporary.

- Network address translation (NAT)

→ dynamically assigned + address translation

→ private vs. public IP address

→ private: Internet routers discard them

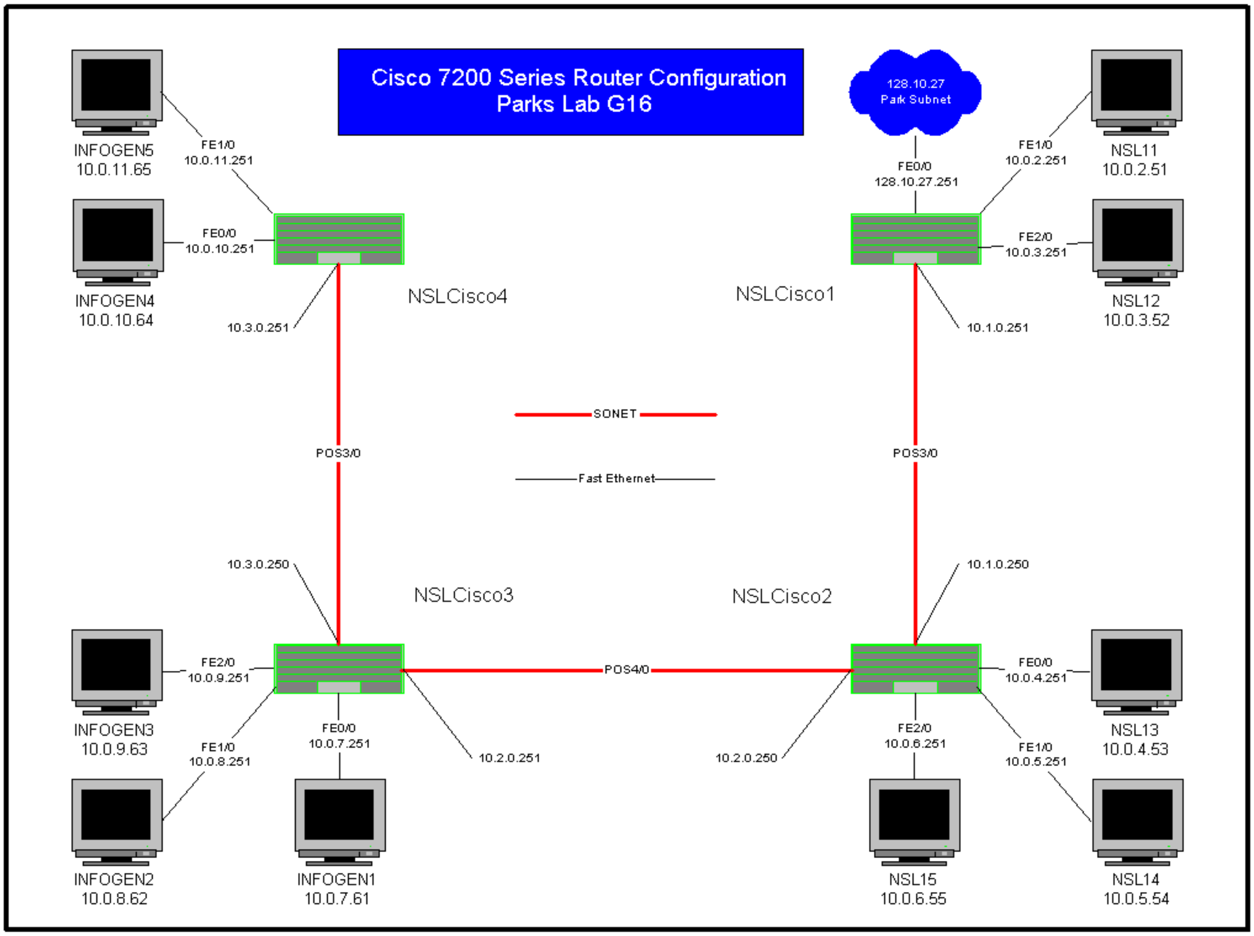
→ e.g., 192.168.0.0 is private

→ 10.x.x.x are also private

→ useful for home networks, small businesses

→ also industry and university research labs

Example: private intranet



- intranet NICs have 10.0.0.0/24 addresses

  $\rightarrow$ each interface: a separate subnet

- only one of the routers connected to Internet

• NAPT (NAT + port)

→ variant of NAT: borrow src port field as address bits

Ex.: 192.168.10.10 and 192.168.10.11 both map to 128.10.27.10

but

→ 192.168.10.10 maps to 128.10.26.10:6001

→ 192.168.10.11 maps to 128.10.26.10:6002

What about port numbers of 192.168.10.10 and 192.168.10.11?

→ e.g., client process bound to 192.168.10.10:22222

→ e.g., client process bound to 192.168.10.11:33333

Doesn't matter: NAPT translation table entries

→ 192.168.10.10:22222 maps to 128.10.26.10:6001

→ 192.168.10.11:33333 maps to 128.10.26.10:6002

For example:

if 192.168.10.10:22222 is a web browser (say Firefox) downloading web page from www.purdue.edu:80

→ web server knows client as 128.10.27.10:6001

→ no ambiguity or confusion

→ similarly for 192.168.10.11:33333

NAPT yields huge increase in effective IP address space

→ IP address bits are increased to 48 (= 32 + 16)

→ biggest factor preventing IP address depletion

Technical problems with NAPT?

Difficult to run servers behind DHCP intranet:

→ how to discover server's dynamic IP address?

→ how to discover server's dynamic port number?

→ NAT traversal problem

Old solution: pay more to ISP to get fixed public IP address and port number

→ not a good customer solution

→ lots of P2P apps, VoIP, gaming, etc.

Two methods:

1. Proxies/relays

→ e.g., Skype: clients contact well-known server—server knows their dynamic addresses

→ server informs client its peer's dynamic IP address and port number

→ peers can talk to each directly

→ also called UDP/TCP hole punching

2. Enhanced gateway capabilities

→ e.g., IGD (Internet Gateway Device) in UPnP

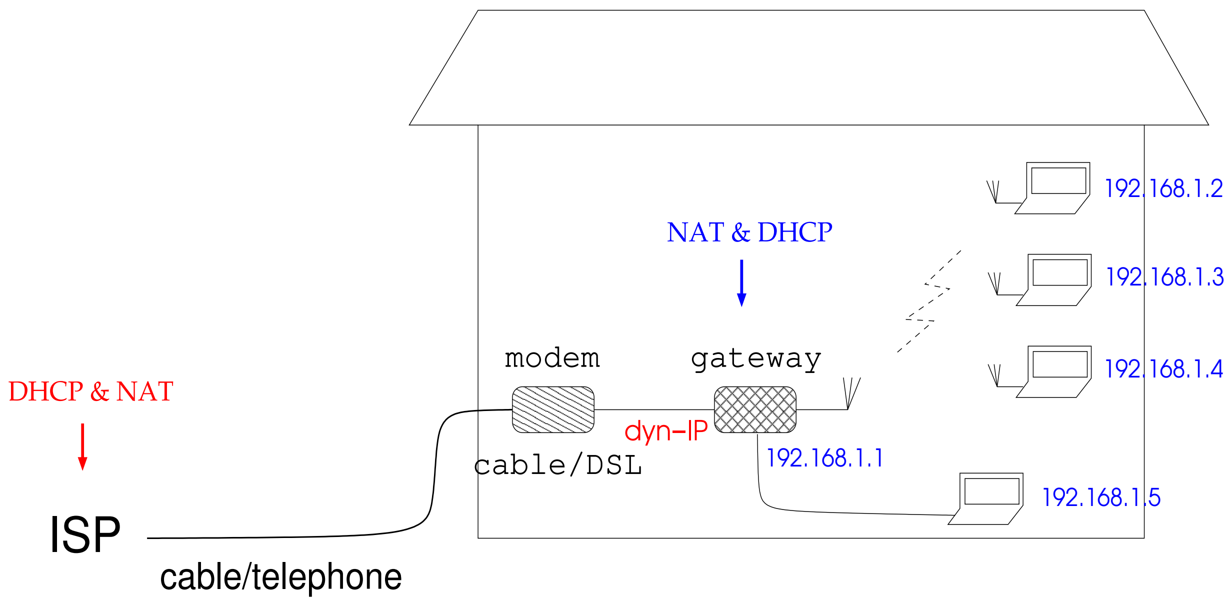
→ IGD compliant router allows user to specify desired port number

→ not much help with dynamic IP address

→ user communicates desired port number via UPnP protocol

Ex.: SOHO (small office/home office)

$\longrightarrow$ now: home networking



- dynamic IP address provided by ISP is shared through NAT
- recall: private IP addresses

  $\rightarrow$ 10.0.0.0/8, 172.16.0.0–172.31.255.255, 192.168.0.0/16

DHCP: 2-phase protocol

1. Discovery

→ client sends broadcast discovery message (UDP, client port 68, server port 67) on LAN

→ one or more DHCP servers respond with dynamic IP address

2. Allocation

→ client sends broadcast message requesting selected IP address

→ DHCP server confirms assignment

DHCP does other network configuration chores:

→ provides DNS server names

→ first-hop router/gateway

→ subnet mask

CIDR and dynamically assigned IP addresses with NAPT

→ significant increase of Internet's effective address space

→ saved the day

Last free IPv4 address block allocated by IANA (suborganization of ICANN) to regional registries early 2011

→ RIRs: ARIN, RIPE, APNIC, LACNIC, AFRINIC

Last available/recovered address pool allocated mid-2014

→ from central Internet authorities to autonomous systems

→ ISPs manage their own address blocks

→ unused address blocks

Back to address space crunch?

→ recurrent push for IPv6

→ ISPs and companies reluctant

→ technical, overhead, and cost issues

→ not backward compatible with IPv4

→ must use separate compatibility mechanisms (e.g., tunneling, hybrid sockets)

→ not-so-pleasant history/memories