



IPv4 Addressing, Subnetting, VLSM and NAT

Networking track | IPv4 header, mask arithmetic, VLSM drills with worked answers, summarization, special-purpose ranges, NAT/PAT types and their security limits | Sources: Tanenbaum, Computer Networks 5th ed. 5.6; RFC 791, 4632, 6890, 4787

#IPv4 #Subnetting #VLSM #NAT

≡ Vocabulary and standards used on this sheet

Prefix / subnet mask — The leading bits that identify the network. Written as a length (/24) or as a dotted mask (255.255.255.0): both say that the first 24 of the 32 bits are the network part, the rest identify the host.

CIDR (Classless Inter-Domain Routing) — Addressing and routing with prefixes of any length instead of fixed classes A/B/C (RFC 4632, 2006; first RFC 1519, 1993). Makes aggregation and VLSM possible.

VLSM (Variable-Length Subnet Masking) — Cutting one block into subnets of different sizes (a /25 for 100 users, a /30 for a router link) instead of equal-size subnets.

Wildcard mask — The bitwise inverse of a mask (0 = must match, 1 = ignore). Used by Cisco ACLs (Access Control Lists) and OSPF network statements.

NAT / PAT — Network Address Translation rewrites IP addresses at a boundary; Port Address Translation (NAT overload) also rewrites ports so many private hosts share one public address.

CGNAT (Carrier-Grade NAT) — NAT run by the ISP for many subscribers at once, using the shared 100.64.0.0/10 space between the customer router and the ISP (RFC 6598). Customer + ISP NAT = NAT444.

Bogon / martian — A source address that must never appear on the public Internet: private, reserved, unallocated or loopback space. 'Martian' is the router term for such packets.

uRPF (unicast Reverse Path Forwarding) — Anti-spoofing check: the router accepts a packet only if its routing table would send replies to that source out of the same interface (strict) or out of any interface (loose).

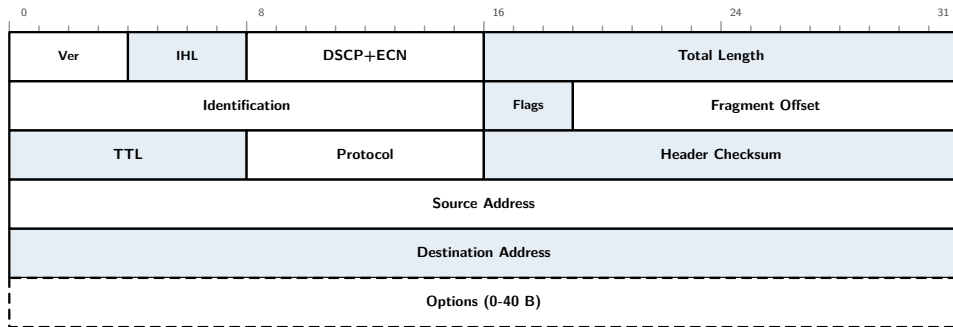
BCP (Best Current Practice) — An IETF document series for operational rules. BCP 38 (RFC 2827) = ingress filtering against spoofed sources; BCP 84 (RFC 3704) = the same for multihomed networks.

ALG (Application Layer Gateway) — NAT helper that rewrites addresses embedded in the payload of protocols such as FTP or SIP (Session Initiation Protocol, VoIP signalling).

RFC 791 (1981)	Internet Protocol: the IPv4 header, fragmentation, TTL
RFC 1918 (1996)	Address Allocation for Private Internets: 10/8, 172.16/12, 192.168/16
RFC 3021 (2000)	Using 31-bit prefixes on point-to-point links (no network or broadcast address)
RFC 4632 (2006)	CIDR: the Internet address assignment and aggregation plan
RFC 4787 (2007)	NAT behavioral requirements for unicast UDP: defines mapping and filtering types
RFC 6302 (2011)	Logging recommendations for Internet-facing servers: log source port + exact time
RFC 6890 (2013)	Special-purpose IP address registries: every reserved IPv4/IPv6 block in one table

🔗 The IPv4 header (RFC 791, Tanenbaum 5.6.1)

Field	Bits	What it carries	Why an analyst cares
Version	4	always 4	6 here means an IPv6 packet
IHL (Internet Header Length)	4	header size in 32-bit words: 5 = 20 bytes, max 15 = 60	IHL > 5 = IP options: rare, often dropped (source routing)
DSCP + ECN	6 + 2	former ToS byte: QoS class (EF = 46 for voice) and Explicit Congestion Notification	unexpected marking from hosts = QoS abuse; rewrite at trust boundary
Total Length	16	header + data in bytes, max 65 535	tells you the packet size against the MTU
Identification	16	same value on every fragment of one datagram	incrementing IDs once allowed idle (zombie) scans
Flags (DF, MF)	3	DF = Don't Fragment, MF = More Fragments follow	DF set + too big = ICMP 'fragmentation needed' (PM-TUD)
Fragment Offset	13	position of this fragment in 8-byte units	overlapping offsets = evasion (Teardrop, 1997)
TTL (Time To Live)	8	decremented by each router; 0 = drop + ICMP Time Exceeded	initial 64 Linux/macOS, 128 Windows, 255 network gear: passive OS fingerprinting
Protocol	8	next header: 1 ICMP, 6 TCP, 17 UDP, 47 GRE, 50 ESP, 51 AH, 89 OSPF	odd values (41, 47) = tunnels past the firewall
Header Checksum	16	covers the header only, re-computed at every hop (TTL changes)	no integrity for the data: that is TCP/UDP's job
Source / Destination Address	32 + 32	the two IPv4 addresses	the source is never authenticated: spoofing, hence BCP 38
Options	0-320	record route, timestamp, loose/strict source route	source routing lets the sender choose the path: drop it



The same header drawn to scale, 32 bits per row: minimum 20 B (IHL 5), maximum 60 B (IHL 15), datagram up to 65 535 B, and every host must accept at least 576 B

Mask arithmetic you must do in your head

Bits in octet	Octet value	Block size	Prefixes that end here
1	128	128	/1 /9 /17 /25
2	192	64	/2 /10 /18 /26
3	224	32	/3 /11 /19 /27
4	240	16	/4 /12 /20 /28
5	248	8	/5 /13 /21 /29
6	252	4	/6 /14 /22 /30
7	254	2	/7 /15 /23 /31
8	255	1	/8 /16 /24 /32

Quantity	Formula	Note
Addresses in a /p	2^{32-p}	a /20 holds 4 096
Usable hosts	$2^{32-p} - 2$	minus network and broadcast; except /31 (2 usable, RFC 3021) and /32 (1 host route)
Subnets from borrowing b bits	2^b	the first ('subnet zero') and last subnets are usable: Cisco IOS enables ip subnet-zero by default since 12.0; the old $2^b - 2$ rule is obsolete
Block size	$256 - m$	m = mask value in the interesting octet
Wildcard (ACL, OSPF)	$255.255.255.255 - \text{mask}$	/21 → 0.0.7.255
Bits to aggregate n blocks	$\lceil \log_2 n \rceil$	bits needed to cover n contiguous blocks

- Find the interesting octet** the octet where the mask is neither 255 nor 0 (/19 → third octet, mask 224)
- Block size = 256 - mask value** 224 → 32; subnets start at 0, 32, 64, 96, ...
- Network = largest multiple of the block** ≤ the address octet octets to the right become 0

4 Broadcast = next network - 1 octets to the right become 255

5 Usable = network + 1 to broadcast - 1 first usable is the conventional gateway

The magic-number method (works for any prefix in 20 seconds)

Drill	Network	Broadcast	Usable range / answer
10.1.77.200/19	10.1.64.0	10.1.95.255	.64.1 - .95.254 (8 190 hosts)
172.31.200.10/21	172.31.200.0	172.31.207.255	.200.1 - .207.254 (2 046)
192.168.5.130/26	192.168.5.128	192.168.5.191	.129 - .190 (62)
192.168.5.94/27	192.168.5.64	192.168.5.95	.94 is the last usable host
192.168.1.47/28	192.168.1.32	192.168.1.47	trap: .47 is the broadcast
10.0.0.255/23	10.0.0.0	10.0.1.255	trap: .0.255 is a valid host
172.16.0.0/16 → /27	-	-	$2^{11} = 2\ 048$ subnets of 30 hosts
Need 500 hosts/subnet	-	-	/23 (510); /24 gives only 254
Need 6 subnets from a /24	-	-	borrow 3 → /27, 30 hosts each
How many /30 in a /26?	-	-	$2^{(30-26)} = 16$

PROBLEM Which subnet holds 192.168.5.130/26, and what is its broadcast?

- GIVEN**
- Mask /26 = 255.255.255.192
 - Only the last octet is split

1 Write the host octet in binary

130 = 1000 0010; mask 192 = 1100 0000

Why: a /26 keeps 2 bits of the last octet as network

2 AND with the mask

1000 0010 AND 1100 0000 = 1000 0000 = 128

Why: network bits survive, host bits clear

3 Set the 6 host bits to 1

1011 1111 = 191

Why: all-ones host part = broadcast

ANSWER Network 192.168.5.128/26, broadcast 192.168.5.191, hosts .129-.190 (62 usable)

CHECK ✓ Block size 256 - 192 = 64: blocks start at .0, .64, .128, .192, and 130 falls in the .128 block

EXAM TIP

Exams hide the trap in the host octet: an address ending in .0 or .255 can be a valid host (10.0.0.255/23), and an address ending in .47, .63, .127 can be a broadcast (/28, /26, /25). Always compute the block before judging. A /31 on a point-to-point link (RFC 3021) has two usable addresses and no broadcast; Cisco IOS accepts it with a warning.

Self-test: a colleague's subnetting notes

There are 7 mistakes. Underline and correct them.

A /26 mask is 255.255.255.192 and leaves 64 usable hosts per subnet.

Host 10.1.1.70/27 sits in subnet 10.1.1.64, whose broadcast address is 10.1.1.127.

A /30 point-to-point link offers 4 usable addresses; RFC 3021 lets you use a /29 there instead, with no network or broadcast address.

The four networks 172.20.0.0/16 to 172.23.0.0/16 summarise to 172.20.0.0/12.

Addresses handed out behind an ISP's carrier-grade NAT come from 192.0.2.0/24.

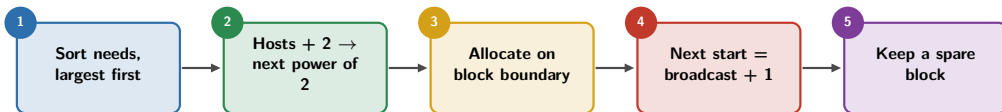
Seven planted errors, struck and corrected. Tells: usable hosts = $2^{\text{host bits}} - 2$; the broadcast is the next block boundary minus 1; RFC 3021 allows /31 on point-to-point links; 100.64.0.0/10 is carrier-grade NAT space (RFC 6598), 192.0.2.0/24 is documentation space

🕒 Classful legacy (still asked, no longer used for routing)

Class	First octet	Leading bits	Default mask	Status
A	1-126 (127 = loopback)	0	/8 (255.0.0.0)	obsolete
B	128-191	10	/16 (255.255.0.0)	obsolete
C	192-223	110	/24 (255.255.255.0)	obsolete
D	224-239	1110	none (group addresses)	multicast
E	240-255	1111	none	reserved

Classes fixed the mask from the first bits of the address, so an organisation got 256, 65 536 or 16.7 million addresses and nothing in between. Class B exhaustion in the early 1990s led to CIDR (1993). Today a router ignores the class: only the configured prefix length counts. Exams still use 'classful boundary' for auto-summarization (RIPv1, EIGRP before IOS 15).

☰ VLSM drills with worked answers



📄 Drill 1: 192.168.10.0/24 for Sales 100, Eng 50, Mgmt 20, Servers 10, three router links

- Sales 100 → needs 102 → 128 block → **192.168.10.0/25** (.1 - .126, bcast .127)
- Eng 50 → 52 → 64 block → **192.168.10.128/26** (.129 - .190, bcast .191)
- Mgmt 20 → 22 → 32 block → **192.168.10.192/27** (.193 - .222, bcast .223)
- Servers 10 → 12 → 16 block → **192.168.10.224/28** (.225 - .238, bcast .239)
- Links → **.240/30, .244/30, .248/30**; spare .252/30. With /31: .240/31, .242/31, .244/31 and .246 - .255 spare

📄 Drill 2: 10.20.0.0/22 for 400, 200 and 100 hosts plus two WAN links

- 400 → 402 → 512 block → **10.20.0.0/23** (10.20.0.1 - 10.20.1.254)
- 200 → 202 → 256 block → **10.20.2.0/24** (10.20.2.1 - 10.20.2.254)
- 100 → 102 → 128 block → **10.20.3.0/25** (10.20.3.1 - 10.20.3.126)
- WAN (Wide Area Network) links → **10.20.3.128/30** and **10.20.3.132/30**; free: 10.20.3.136 - 10.20.3.255 (a /29, a /28, a /26 remain)

10.40.0.0/20 = 4 096 addresses · 89.1 % allocated



Drill 3, a cloud VPC (Virtual Private Cloud) 10.40.0.0/20 over three availability zones (AZs, separate data centres of one region): largest blocks first, one app, public and data subnet per AZ, same size in every AZ so a failover lands on identical capacity. The top 448 addresses stay free as aligned blocks for a fourth AZ or growth. Usable counts subtract the 5 addresses AWS and Azure keep in every subnet (network, router, DNS, reserved, broadcast)

⚠️ Why smallest-first breaks VLSM

Allocating a /28 at .0 and then trying to place a /25 forces it to .128, wasting .16 - .127 or overlapping. Overlapping subnets on two router interfaces are refused by IOS ('overlaps with ...'); on firewalls and cloud VPCs (Virtual Private Clouds) they silently break routing and peering. Leave room to grow each block to the next prefix.

☞ Summarization (route aggregation)

- Write the varying octet of the first and last network in binary
- Count the leading bits they share; prefix = bits of fixed octets + shared bits
- Summary network = first network with the non-shared bits zeroed
- Check that the summary does not cover networks that live elsewhere

Find the summary of a set of contiguous networks

📄 172.16.8.0/24 to 172.16.15.0/24 → one route

- 8 = 0000 1000, 15 = 0000 1111: the first 5 bits match → 16 + 5 = /21
- Summary **172.16.8.0/21** (255.255.248.0) covers exactly .8 - .15, eight /24s, nothing more
- Counter-example: 10.1.5.0/24 - 10.1.8.0/24 (5 = 0000 0101, 8 = 0000 1000) share 4 bits → 10.1.0.0/20 covers .0 - .15: over-summarization, attracts traffic for twelve networks you do not own

Where summaries are configured (Cisco IOS)

```
! OSPF: on the ABR (Area Border Router), per area
router ospf 1
 area 1 range 172.16.8.0 255.255.248.0
! EIGRP: per interface
interface GigabitEthernet0/1
 ip summary-address eigrp 100 172.16.8.0 255.255.248.0
! BGP: announce the aggregate only
router bgp 65010
 aggregate-address 172.16.8.0 255.255.248.0 summary-only
! Static discard route against loops for unused space
ip route 172.16.8.0 255.255.248.0 Null0
```

Longest prefix match — The router picks the most specific matching route. With 10.0.0.0/8, 10.1.0.0/16, 10.1.1.0/24 and 0.0.0.0/0 in the table, 10.1.1.9 follows the /24 and 10.1.2.9 the /16, whatever the administrative distance (AD, the trust rank of the route source). AD only breaks ties between identical prefixes.

Summary black hole — The summarizing router installs a route to Null0 (a discard interface); if one component subnet fails, traffic is dropped there instead of looping. Intentional, but hides partial outages from upstream.

Hijack by more-specific — An attacker announcing a longer prefix than your summary wins every router's longest-match decision. Announce /24s for critical space and publish ROAs (Route Origin Authorizations, signed in the RPKI, Resource Public Key Infrastructure) with a maxLength so routers doing ROV (Route Origin Validation, RFC 6811) reject the bogus origin.

🛡️ Special-purpose IPv4 ranges (RFC 6890 registry) and bogon filtering

Block	Purpose	Reference	At the Internet edge
0.0.0.0/8	'this network', DHCP source 0.0.0.0	RFC 791/1122	drop
10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16	private use	RFC 1918	drop inbound, never route out
100.64.0.0/10	shared CGNAT space	RFC 6598	drop (unless you are the ISP)
127.0.0.0/8	loopback	RFC 1122	drop (martian)
169.254.0.0/16	link-local; APIPA (Automatic Private IP Addressing, used when DHCP fails); cloud meta-data 169.254.169.254	RFC 3927	drop
192.0.0.0/24	IETF protocol assignments (192.0.0.8/32 ICMP dummy)	RFC 6890	drop
192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24	documentation TEST-NET-1/2/3	RFC 5737	drop
192.88.99.0/24	6to4 relay anycast, deprecated	RFC 7526	drop

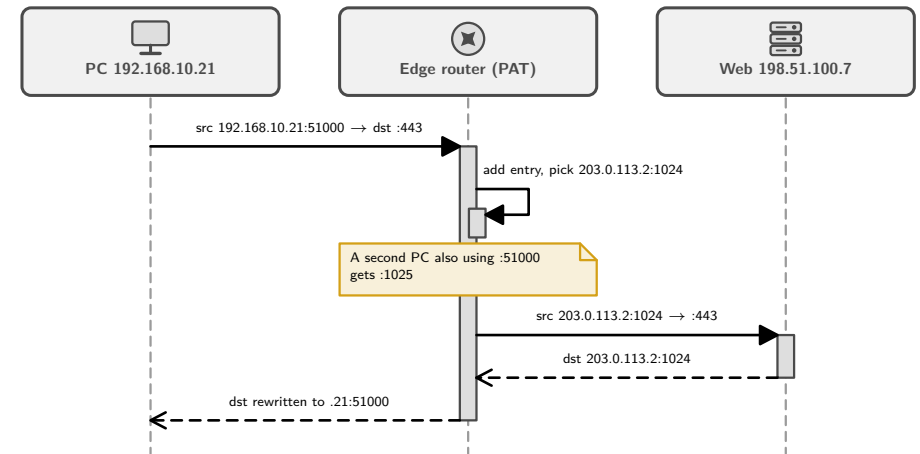
198.18.0.0/15	benchmarking (lab testers)	RFC 2544	drop
224.0.0.0/4	multicast (224.0.0.0/24 link-local, never routed)	RFC 5771	drop unless multicast peering
240.0.0.0/4, 255.255.255.255/32	reserved (class E), limited broadcast	RFC 1112 / 919	drop

Anti-spoofing at the edge

✓ baseline / mandatory ❑ recommended – verify in your context

- ✓ **Ingress**: drop bogon and your own prefixes as source on upstream links BCP 38 / RFC 2827: a packet arriving from the Internet claiming to come from inside is forged
- ✓ **Egress**: permit only your allocated prefixes as source stops your hosts joining reflection DDoS (Distributed Denial of Service) with forged sources
- ❑ **MUST** Strict uRPF on single-homed customer edges `ip verify unicast source reachable-via rx` (BCP 84 / RFC 3704)
- ❑ **SHOULD** Loose uRPF (`reachable-via any`) on multihomed links strict mode drops legitimate asymmetric traffic (reply returns on another link)
- ❑ **COULD** Automate full-bogon lists (unallocated space changes) Team Cymru bogon feed or RPKI-based filters

→ NAT and PAT: types, commands, security limits



PAT (NAT overload) on one flow. Cisco terms: inside local = the host's real private address, inside global = the public address it appears as, outside global = the real remote address. The router keeps one table entry per 5-tuple and rewrites the source port when two hosts pick the same one

Inside local	the real (private) address of the inside host, as configured on it
Inside global	the public address the inside host appears as on the Internet
Outside global	the real address of the remote host (the web server)
Outside local	how the remote host appears to inside hosts; equals outside global unless the destination is also translated

Static NAT	Dynamic NAT	PAT / overload	Port forwarding
1 private ↔ 1 public, permanent	private ↔ first free public from a pool	many private → 1 public, unique source ports	1 public IP:port → 1 private IP:port
inbound possible	inbound only while mapped	outbound only	inbound to one service
published servers	rare today (pool exhaustion)	every home and branch	static PAT, DNAT (destination NAT)

```
Cisco IOS: PAT, static PAT, pool, verification
interface GigabitEthernet0/0
 ip nat inside
interface GigabitEthernet0/1
 ip nat outside
access-list 1 permit 192.168.10.0 0.0.0.255
ip nat inside source list 1 interface Gi0/1 overload
ip nat inside source static tcp 192.168.10.5 443 203.0.113.10 443
ip nat pool P00L 203.0.113.20 203.0.113.30 prefix-length 27
ip nat inside source list 1 pool P00L
show ip nat translations
show ip nat statistics
clear ip nat translation *
```

Linux / Windows	Effect
sysctl -w net.ipv4.ip_forward=1	Enable routing (persist in /etc/sysctl.d/)
nft add rule ip nat postrouting oifname eth0 masquerade	PAT to the egress address (nftables: table ip nat, chain postrouting type nat)
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE	Legacy iptables equivalent
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to-destination 192.168.10.5:443	Port forwarding; also needs a FORWARD accept rule
conntrack -L -p tcp --dport 443	Live translation and state table (conntrack-tools)
ipcalc 10.1.77.200/19 ; sipcalc 10.1.77.200/19	Verify subnet math from the shell
Get-NetIPAddress -AddressFamily IPv4 ft IPAddress,PrefixLength	Windows addresses and prefixes
New-NetNat -Name lab -InternalIPInterface AddressPrefix 192.168.50.0/24	Windows host NAT (Hyper-V labs)

RFC 4787 behaviour	Old 'cone' name	Effect on P2P / security
Endpoint-independent mapping (EIM) + filtering	full cone	one public port per inside socket whatever the destination; any host can reach it: best for P2P (peer-to-peer), worst exposure
EIM + address-dependent filtering	restricted cone	only hosts you contacted may reply
EIM + address-and-port-dependent filtering	port-restricted cone	reply only from the exact IP:port
Address-and-port-dependent mapping	symmetric	new mapping per destination; STUN (Session Traversal Utilities for NAT, discovers your public mapping) fails, a TURN relay (Traversal Using Relays around NAT) is needed

Is NAT a security control?

✓ What it gives

+ Unsolicited inbound has no mapping, so it is dropped (a side effect of state)

+ Hides internal addressing from casual observers

+ Conserves IPv4 space (CGNAT, RFC 6598)

✗ What it does not

- No inspection: outbound C2 (command-and-control), phishing and exfiltration pass freely

- ALGs rewriting SIP/FTP payloads can be abused (NAT Slipstreaming, 2020: a web page tricks the ALG into opening an inbound port): disable unused ALGs

- CGNAT breaks attribution unless source port + timestamp are logged (RFC 6302)

- Port forwarding exposes the internal host exactly like a public IP

Addressing mistakes seen in audits and their consequence

- **Overlapping RFC 1918 ranges between sites, partners or cloud VPCs** VPNs and peering need NAT hacks; plan a global allocation first (IPAM)
- **One flat /16 for a whole site** huge broadcast domain, no filtering point between user groups
- **192.168.0.0/24 or 192.168.1.0/24 used internally** collides with home routers: remote-access users cannot reach those subnets through the VPN
- **Gateway always at .1, documented per subnet**
- **/31 or /30 links and /32 loopbacks from a dedicated infrastructure block** easy ACLs and summarization for infrastructure addresses

Q1

Network, broadcast and usable range of 10.1.77.200/19?

A

10.1.64.0, 10.1.95.255, 10.1.64.1 - 10.1.95.254 (block 32 in the third octet).

Q2

Summary route for 172.16.8.0/24 through 172.16.15.0/24?

A

172.16.8.0/21 (255.255.248.0).

Q3

Which NAT term names the public address a private host appears as?

A

Inside global (inside local = its real private address).

Q4

Why must a CGNAT operator log source ports?

A

Thousands of subscribers share one public IP; only IP + port + time identifies one (RFC 6302).

Q5

Wildcard mask for a /27 in an ACL?

A

0.0.0.31.

Q6

Which ranges should never appear as a source on your Internet link, besides RFC 1918 private space?

A

Any bogon: 0/8, 100.64/10, 127/8, 169.254/16, 192.0.2/24, 198.18/15, 224/4, 240/4, and your own prefixes (BCP 38 ingress filtering).

Q7

An IPv4 header shows IHL = 6. What does it mean?

A

24-byte header: 4 bytes of IP options follow the fixed 20 bytes; unusual, inspect it.

Q8

TTL 117 arrives at your sensor. Likely sender OS?

A

Windows (initial 128) about 11 hops away; Linux starts at 64, network devices at 255.