IP Networking and Routing

John W0VG
Willem AC0KQ

IPv4 Network Definition

- Packet switched network
  - Introduced in 1981
  - Replaced older circuit switched networks
- Advantages
  - Decentralized, hierarchical configuration
  - Robust, dynamic routing
  - Supports 4,294,967,296 unique addresses
- Every device must have a unique IP address
- Network of networks (subnets)

RMHAM NetLab

- Subset of mountain top sites
  - Purple wires are microwave links
  - Red wires are commercial internet (VPN) links
  - Blue wires are local (LAN) links
- Configured to operate like the real network
  - Same IP addresses but blank passwords
  - OSPF for routing
  - “The Internet” is a single router + fancy routing
- Used for training and testing by NetOps

Mikrotik Hardware

- RouterOS is designed for routing
  - Custom Linux kernel
  - Supports IPv4, OSPF, BGP, …
  - Any port can perform any function
  - Same interface regardless of model
  - Command line and well designed GUI
- Reasonable cost
  - Very reliable (but lightning...)
  - Targets Wireless ISP market
    - Long range radios (2GHz, 5GHz, 60GHz)
    - International versions can operate in Ham Band

Classic IP tools

- ping
  - Tests end-to-end connectivity
- traceroute
  - Shows the route the packet takes
  - Windoze calls it **tracert**
- ifconfig
  - Show local IP configuration
  - Windoze calls it **ipconfig**
- route
  - Displays routes

Other versions of IP

- IPv1, IPv2, IPv3
  - Early development versions
- IPv4
  - What the internet is mostly based on
- IPv5
  - Experimental Quality of Service addition
- IPv6
  - Next generation IP (128 bit addresses)
  - New and improved!

Uses of IP

- Most successful protocol ever
  - Ubiquitous internet spans the world
  - Numerous private/isolated versions
    - RMHAM
    - Phone companies
    - Department of Defence
    - NASA Deep Space Network (with special mods)
- Applications
  - Data (HTTP, SMTP, FTP, ...)
  - Voice Over IP (VoIP)
  - Video streams (RTSP etc)
  - Internet of Things (IoT)

IP Addresses

- Basic building block to identify end points
- 32 bits organized as 4 octets
  - 10.30.20.6
  - 00001010 00011110 00010100 00000110
    - subnet part (24 bits)
    - host part (8 bits)
  - All 32 bits are used to identify the device
  - The subnet part determines what can be reached **locally**
  - This can also be written as 10.30.20.6/24
- IPv6 expands this to 128 bits

What is a subnet?

- Part of a greater (interconnected) network
- Group of IP addresses that can directly communicate (e.g. via ethernet or WiFi)
- Sometimes called Local Area Network (LAN)
- Devices on a subnet
  - Have the same leading bits (subnet address)
  - Can directly talk to other devices on the subnet
- Hardware that facilitate device to device communications on a subnet is a hub or switch

Reserved Subnet Addresses

- Network address (host part all zeroes)
  - 10.30.20.0/24
- Broadcast address (host part all ones)
  - 10.30.20.255/24
- Every subnet has these reserved addresses
  - Except /32 and /31
- Other *special* addresses are just by convention
  - Example: gateway often the lowest address
    - 10.30.20.1/24

Bridging and Routing

- Only hosts on the same subnet can be reached directly
  - Subnet part must be the same
- Hosts with a different subnet part must be routed
  - Packets are sent to a special device called a router
  - The router figures out what to do
- A bridge is a smart switch/stupid router that connects two parts of a subnet
  - Must be the same subnet
  - The bridge knows about ***all*** devices on the subnet

Sidebar: Ethernet

- Introduced in 1980 and meshed well with IP
  - Standardized in 1983
- Most commonly used local connections
  - Started at 10Mbps, approaching 1 Tbps
  - Coax, twisted pair (copper) and fiber media
  - Beat out token ring, FDDI, and other technologies
- Ethernet hubs and switches extend network
  - Spanning Tree Protocol (STP) resolves loops
  - No user configuration required
    - All devices are bridged

Terminology

- IP Address
  - The unique 32 bit address for a device
- Netmask
  - A bit mask indicating the subnet part
    - 255.255.255.0 or /24 or 11111111 11111111 11111111 00000000
- Slash notation
  - Shorthand combining IP with netmask
    - 10.30.20.6/24
- Gateway
  - Where to send packets that do not match our subnet
  - MUST be a device on our subnet

Domain Name System (DNS)

- Not required for IP to function
  - Provides symbolic names for humans to use
- Symbolic names for IP addresses
  - router.thorodin.rmham = 10.30.20.1
  - radius.thorodin.rmham = 10.30.20.6
- DNS server provides translation
  - Must be an IP address (can be more than one)
  - Does not have to be on our subnet
  - Google DNS 8.8.8.8, 8.8.4.4
- DNS names are read right to left

How DNS works

- Server sends a UDP packet to DNS server
  - What is the IP address for www.rmham.org?
- Server sends a UDP packet to:
  - root server (IP address for www.rmham.org?)
    - UDP reply: Ask 192.19.56.1 (.org root)
  - .org root (IP address for www.rmham.org?)
    - UDP reply: Ask 162.159.24.80 (ns1.bluehost.com)
  - ns1.bluehost.com (IP address for www.rmham.org?)
    - UDP replay: www.rmham.org = 23.237.17.75
- Server UDP reply to host
  - www.rmham.org = 23.237.17.75

Setting the IP Address Manually

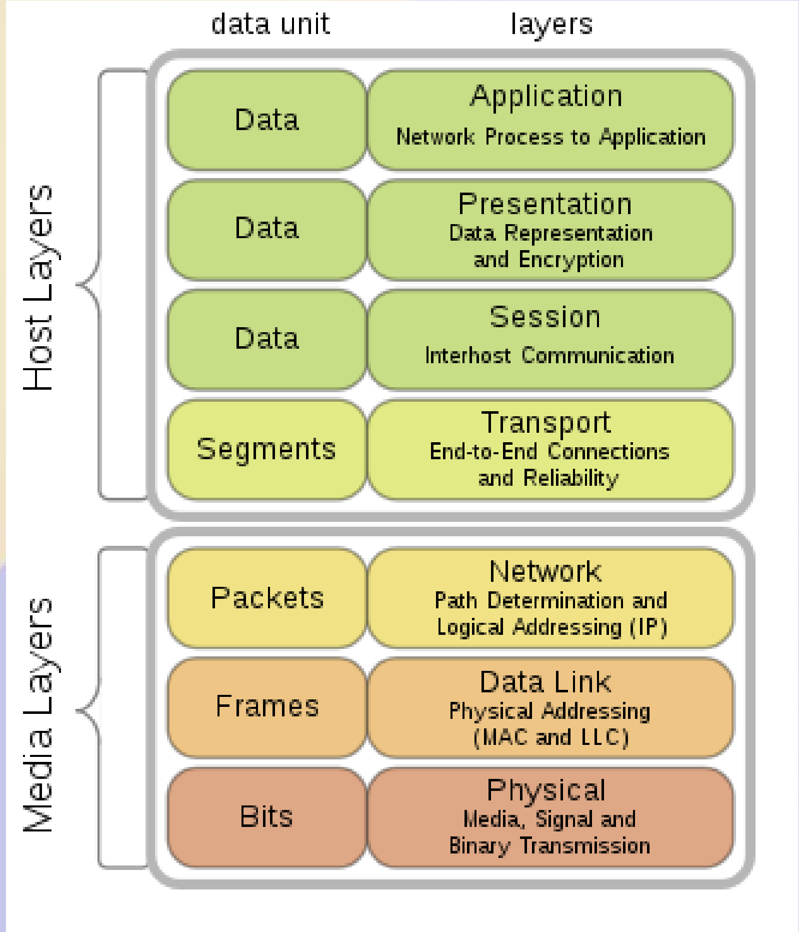
- Determine your subnet address and netmask
  - Example: 10.30.20.0/24
- Select an unused address within that netmask
  - Example: 10.30.20.6
  - Duplicate IP addresses causes havoc
- Determine the gateway address
  - MUST be on our subnet
    - LAN address of router
  - Often first address on subnet (10.30.20.1)
- DNS server (optional)
  - Can be anywhere, often 8.8.8.8

Sidebar: Protocols

- Common language between devices
  - Exchange specific type of information
  - Builds on other protocols
  - Can be specialized packets
- Example data protocols
  - ARP (address resolution on LAN)
  - ICMP (routing error information)
  - TCP/IP (generic virtual switched circuit)
  - HTTP (browser-server built using TCP/IP)

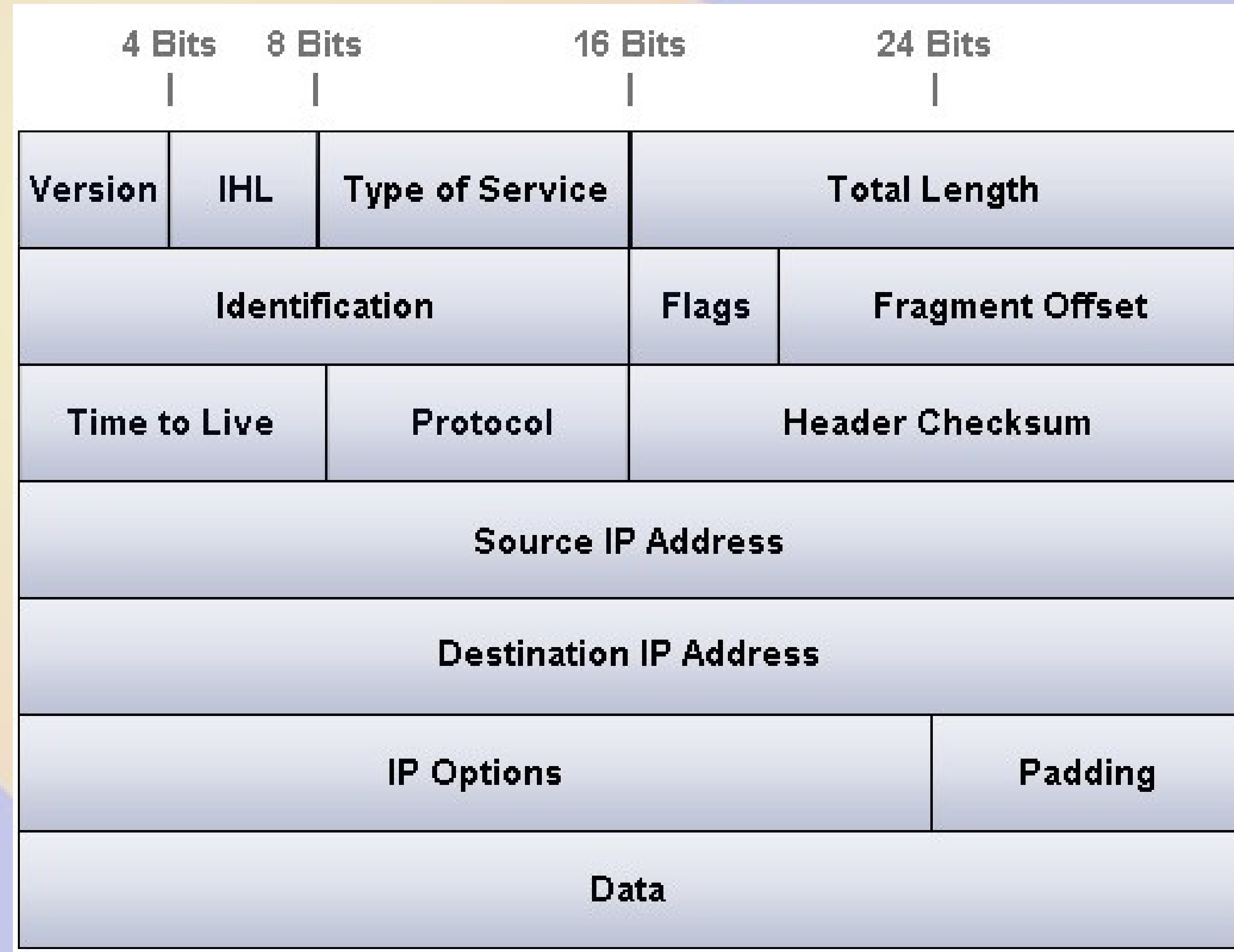
OSI Network Model

- Conceptual representation of network protocols
  - HTTP - Application
  - TCP - Transport
  - IP - Network
  - Ethernet – Data Link
  - IEEE 802.3u – Physical
- Upper layers hide complexity of lower layers



Anatomy of an IP packet

- Version=4
- IHL=IP Hdr Len
- Type of Service
  - Min delay
  - Max throughput
  - etc.
- Flags & Frag Off
  - Large packets



- Protocols add additional header in data section

Internet Control Message Protocol (ICMP)

- Used for debugging and error messages
- Replies generated from IP stack (OS)
  - Unreachable
  - TTL exceeded
  - Redirect
- Used by ping and traceroute

Anatomy of an ICMP Packet

- Typically 8-16 bytes in addition to IP header

IP Datagram

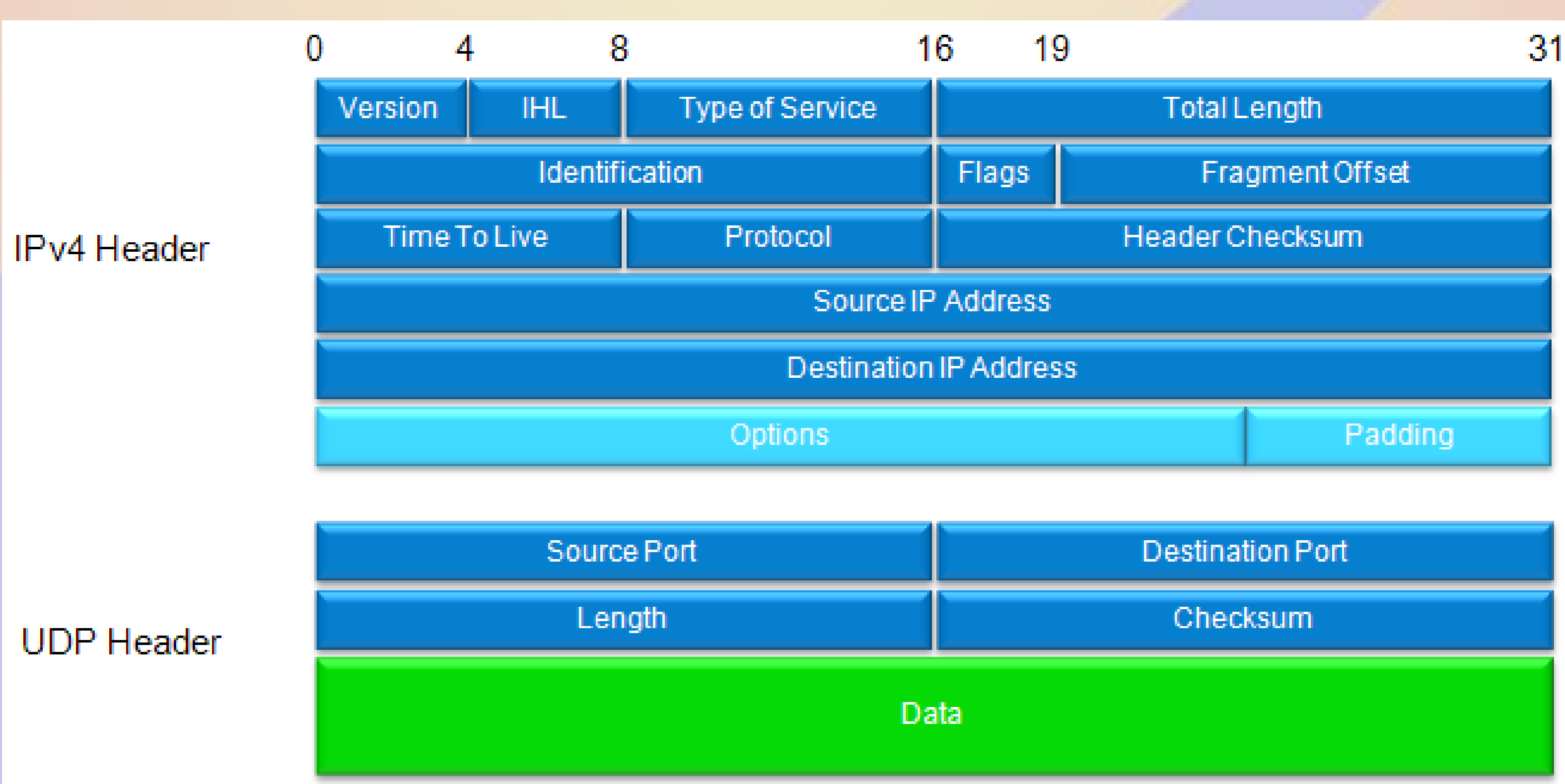
	Bits 0–7	**Bits 8–15**	**Bits 16–23**	**Bits 24–31**
IP Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		*flags* and *offset*	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (*optional*)	Payload Data			

User Datagram Protocol (UDP)

- Single packet datagrams
  - Works well for smaller payloads
- Best effort delivery
  - Unordered arrival
  - Optional checksum
- Used by many higher level protocols
  - DNS, NTP, NFS, ...
  - Software Defined Networking (SDN)
  - VOIP and repeater linking

Anatomy of a UDP Packet

- Adds only port numbers, length and checksum
- One packet contains the complete message
  - Works well for DNS, NTP, etc.



Transmission Control Protocol (TCP)

- Bidirectional virtual circuit
  - Input = Output
  - Traffic arrives in order
  - Retransmit, congestion,etc happens invisibly
  - No inherent framing
- Used by many higher level protocols
  - Terminal connections (ssh, telnet)
  - Data transfer (HTTP, FTP, scp, rsync)

Dynamic Host Configuration Protocol (DHCP)

- Server (router) assigns unique IP address
  - Single point administration
  - Assigns IP address from pool
    - Can do IP reservation by MAC address
- Specialized IP packets to configure host
    - Client sends **broadcast** packet to discover server
    - Server sends **unicast** offer to provide configuration
    - Client sends accept request
    - Server acknowledges acceptance
- Always sets IP address
- Usually sets netmask, gateway, DNS server

Demo site: LabDemo

- ether1 – rest of the network
- LAN (bridge1 = ether2 - ether5)
  - Address: 10.30.80.1
  - Subnet: 10.30.80.0/24
  - DHCP: 10.30.80.200-250
- Routing table
  - 10.30.80.0/24: send to bridge1
  - anything else to be determined

Address Resolution Protocol (ARP)

- Translates IP address to MAC address
  - Needed for ethernet to work (uses MAC addresses)
  - Works only on LAN
    - same wifi or ethernet network
- Media Access Control (MAC)
  - 48 bit address
  - Address scheme for IEEE 802
    - ethernet, WiFi, Bluetooth, ...
- Generally requires no user configuration
  - ARP just does the right thing for us automatically

How Routing Works

- Forward a packet ***towards*** its final destination
  - Routing is done one hop at a time
    - What port (neighbor) do you send it to?
    - Based on destination IP address
- Home or edge routers use the default gateway
  - Anything the router doesn’t know goes here
  - Typically just one connection to the internet
- Multi-port routers select the “best” way
  - Routing tables determine next hop
    - Static routing
    - OSPF intra-domain routing
    - BGP inter-domain routing
    - RIP, IS-IS, EIGRP, EGP, etc.

Classfull Inter-Domain Routing

- Prior to 1993 routers assumed classes
  - Class A = 1.x.x.x to 126.x.x.x
    - 126 nets of 16,777,326 hosts each
  - Class B = 128.0.x.x 191.255.x.x
    - 16,384 nets of 65,536 hosts each
  - Class C = 192.0.0.x – 223.255.255.x
    - 2,752,512 nets of 256 hosts each
  - Class D = 224.x.x.x-239.x.x.x
    - Multicast
  - Special cases 0.x.x.x, 127.x.x.x, etc

Classless Inter-Domain Routing (CIDR)

- Breaks network/host at any bit position
- Example: 10.30.20.0/24
  - Class A address (10.x.x.x)
  - /24 makes it a class C (256 addresses)
- Example: 10.30.20.96/29
  - No classed equivalent
  - 8 addresses
  - Netmask 11111111 11111111 11111111 11111000 = 255.255.255.248
- Extended the life of IPv4
  - Many more unique networks

Special Subnets/Addresses

- 0.0.0.0 Any IP address
- 0.0.0.0/0 Any (or default) network
- 255.255.255.255 Broadcast IP address
- 127.0.0.1 Loopback address
- 10.x.x.x, 192.168.x.x, 172.16-31.x.x Private
  - Not publicly routed but can be internally routed
- 44.x.x.x AMPR (Amateur Packet Radio)

Ports

- 16 bit number 0-65535
- Defines a program to talk to
- Ports 0-1023 are *well-known* ports
  - Defines specific services
    - 22 ssh (secure shell)
    - 25 smtp (mail)
    - 80 http (web)
- Ephemeral ports often used for user programs
- Notation IP:port
  - 10.30.20.6:22 means 10.30.20.6 port 22

Network Address Translation (NAT)

- Also called masquerade
  - Uses port numbers to share an (external) IP
  - Router pretends to be devices behind it
- Router rewrites packets both ways
- Extended the life of IPv4 by decades
- Limited security measure
  - Only the router can be reached from internet
  - Port forwarding allows inbound connections

Connection to the Network with NAT

- This is how most home routers work
  - Most consumer routers can only operate this way
- ether1 = DHCP client
  - Configures network connection
    - address 10.30.31.250/24
    - gateway 10.30.31.1
    - DNS 10.30.20.6
  - Add NAT on ether1
- We present as 10.30.31.250 to the network
  - Router translates to our local IP

Routing with NAT

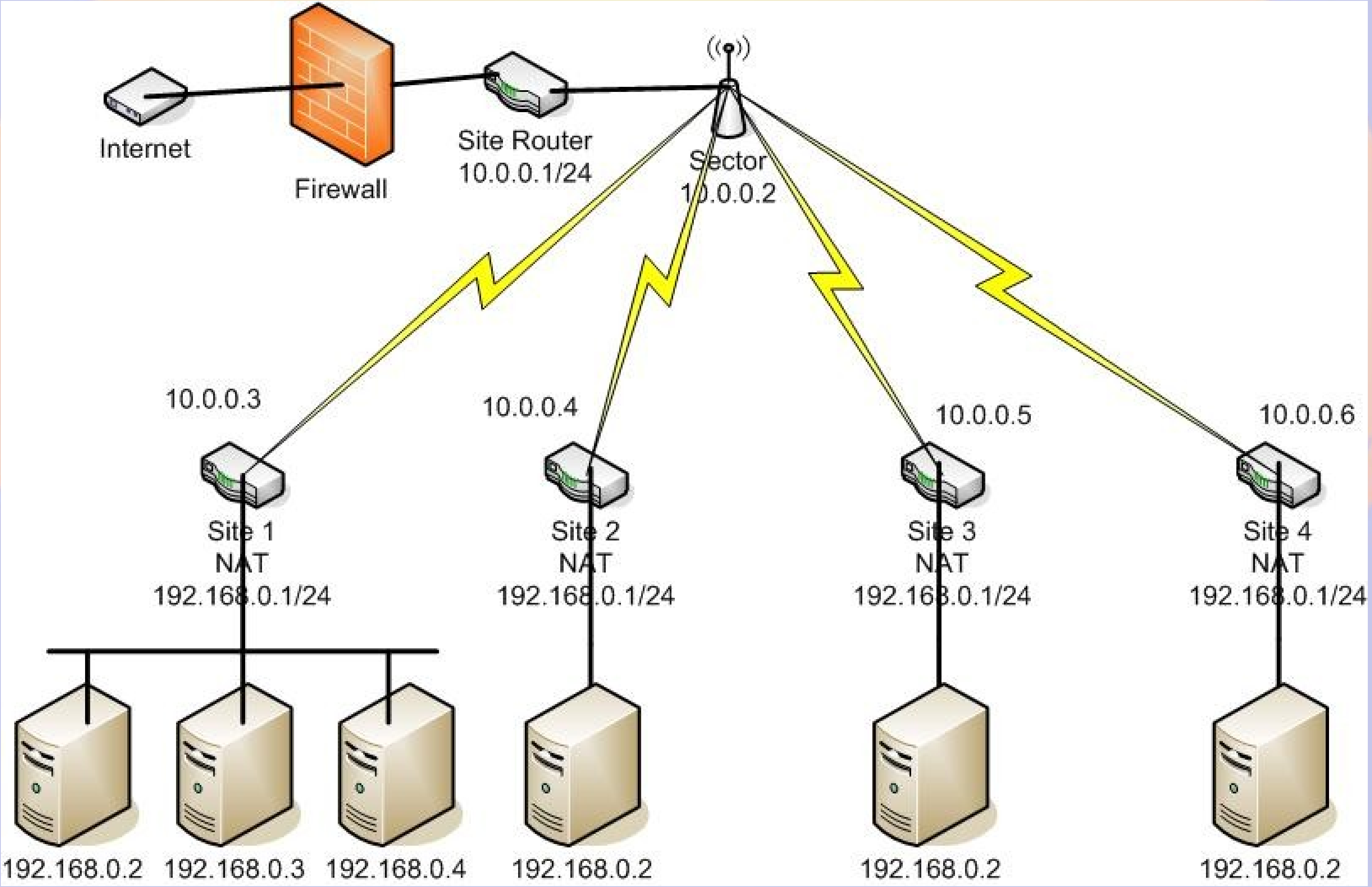
- 10.30.80.x are sent directly via the bridge
- Everything else forwarded with NAT
  - All outbound packets are rewritten to be from our WAN (external) address
    - Our WAN IP must be routable
    - Builds the NAT table
  - All inbound packets are rewritten and forwarded to the right device on the LAN
    - NAT table used to remember where to

Detailed NAT Example

- ssh 10.30.80.200 to 10.30.20.6
  - 10.30.80.200 to 10.30.80.1 via bridge
    - source 10.30.80.200 : 32400, destination 10.30.20.6 : 22
  - Router rewrites packet forwards on network
    - source 10.30.31.1 : 36700, destination 10.30.20.6 : 22
  - Reply returned to router
    - source 10.30.20.6 : 22, destination 10.30.31.1 : 36700
  - Router rewrites reply, forwards to 10.30.80.200
    - source 10.30.20.6 : 22, destination 10.30.80.200 : 324000
- Happens for may packets
  - TCP/IP builds virtual circuit
  - ssh adds encryption

Typical ISP Sector with NAT

- Entire **sector** is one subnet, client site does NAT



Static Routing

- Steer traffic to a specific port
  - route add -net 10.30.20.0/24 dev ether1
  - route add -net 10.30.30.0/24 dev ether2
- Works only for smaller networks and simple cases
  - Every router needs to be configured manually
  - Link failures require lots of manual changes
- Still used to specify special cases
  - Edge routers, laptops, VPNs, etc.
  - Tell OSPF how to get to a special subnet
    - Avoid static routes when using OSPF
    - Misconfigured static routes breaks everything

Route Matching

- Routing is done based on the tightest fit
  - Every bit in the net mask **must** match
  - More bits in netmask => better match
- Example: Destination 10.30.20.6
  - 0.0.0.0/0 loosest (default)
  - 10.0.0.0/8 less loose
  - 10.30.0.0/16 tighter
  - 10.30.20.0/24 tightest
  - 10.30.10.0/24 does not fit (3rd octet does not match)
  - 10.30.20.8/29 does not fit (5 bits in 4th octet mismatch)

Sidebar: Virtual Private Network (VPN)

- Encapsulates IP packets to create a virtual tunnel between devices
  - Virtual circuit at the data link (device) layer
- Can be encrypted for privacy
  - Widely used to defeat censorship
- Very useful for remote access
  - Mobile devices
  - Failover
  - Network integration

Connecting to the Network via VPN

- ether1 = DHCP client
  - Configures network connection
    - address 192.168.1.250/24
    - gateway 192.168.1.1
    - DNS 8.8.8.8
  - Add NAT on ether1
- sstp-thor
  - SSTP VPN tunnel
    - Thorodin 172.16.20.1
    - LabDemo 172.16.20.40

Routing via VPN

- LabDemo Routes
  - 0.0.0.0/0 GW 192.168.1.1
  - 192.168.1.0/24 GW 192.168.1.1
  - 10.0.0.0/8 GW 172.16.20.1
  - 192.168.0.0/16 GW 172.16.20.1
  - 172.16.0.0/16 GW 172.16.20.1
- Only 10.x.x.x.x, 192.168.x.x and 172.16.x.x are routed via the VPN
  - All other traffic goes direct
  - Alternately ALL could be sent to VPN

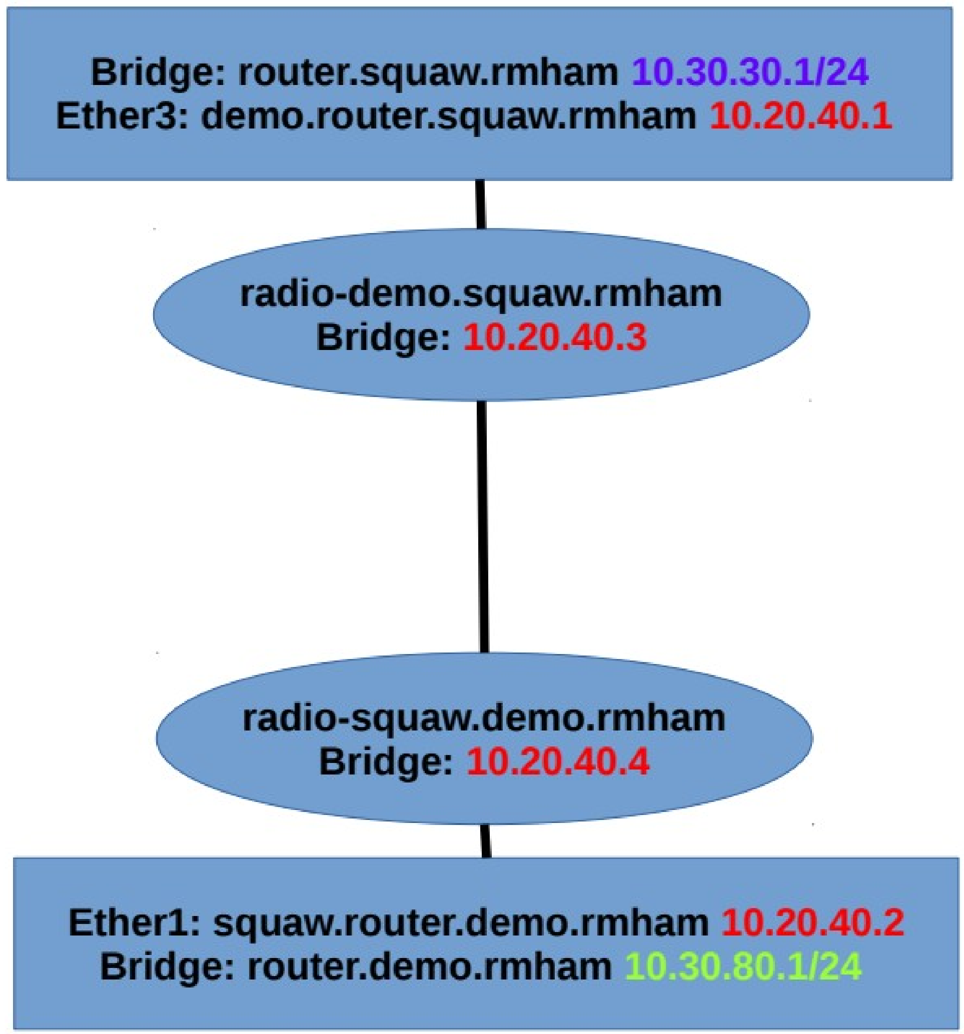
Static Route to Squaw

- Subnet for link from Demo to Squaw: 10.20.40.0/24
- On LabDemo
  - ether1 = 10.20.40.2/24
  - default route gateway 10.20.40.1
  - DNS = 10.30.20.6
  - Routing table
    - 10.30.80.0/24: send to bridge1
    - 10.20.28.0/24: send to ether1
    - everything else: send to 10.20.28.1
- Squaw
  - ether3 = 10.20.40.1/24
  - Static route 10.30.106.0/24: send to 10.20.28.2

Building an Point-to-Point RF link

- The link is a subnet
- The RF link is a bridge
  - 10.20.40.3 is the access point
  - 10.20.40.4 is the client
- Devices are set up to bridge traffic
  - Packets are forwarded to the other end
  - Logically the RF part is transparent

Link Configuration



Route Failover

- 10.20.40.4 (Demo site radio)
- From laptop
  - 10.30.80.X you have to go via the gateway
    - 10.20.40.1 is the gateway
    - 10.30.80.X is only reachable via the gateway
- Add a secondary default gateway
  - gateway=10.20.40.1 distance=1 check=ping
  - gateway=10.20.40.2 distance=2
- Secondary gateway is used when RF link fails

Automatic Routing

- Autonomous Systems (AS)
  - A group of routers using the same protocol
  - RMHAM network is an Autonomous System
- Gateway = Router
- Link = Connection between routers

OSPF Routing

- Open Shortest Path First
- Good for centrally administered networks
  - Fast finding the “best” path
  - Knows about all subnets in domain
  - Works well for <1000 subnets
- Routes traffic via “best” path
  - User defined cost per link
  - Does not do load balancing
- Simple to configure

BGP

- Border Gateway Protocol
- Workhorse of the internet
- Works with very large routing tables
  - Transfer between routers with TCP/IP
  - 1,000,000 routes on internet backbone
  - Based on Autonomous System Number
    - Must be unique to be used publicly
- BGP peer routers form a mesh
  - Determines cheapest route

Configuring OSPF

- Router instance
  - Router ID (use bridge IP address)
  - Redistribute routes
    - Default – never
    - Connected – as type 1
    - Static – as type 1
- Networks you want OSPF to manage
  - RMHAM uses 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/16
- Interface (link) properties
  - Interface type (point-to-point most robust)
  - weights (default=10)
    - Traffic goes via least expensive cumulative weight

Sidebar: Virtual LAN (VLAN)

- Partition a physical network at ethernet level
  - Send ethernet packets as if physical separation
- Applications
  - Security (e.g. DMZ)
  - Storage Area Network (SAN)

Network Management Tools

- cping
  - Concurrent ping
- SmokePing
  - Graphing ping
- Observium/Nagios/LibreNMS/…
  - Network Management Systems
- RANCID
  - Network device backup

Other Network Tools

- nmap
  - network exploration and port scanner
- WireShark
  - ethernet packet sniffer
- WinBox
  - Torch port
  - RoMON
  - mactelnet

Network Time Protocol (NTP)

- Syncs time over network
  - UDP packets with precise time stamps
  - Corrects for network delay both ways
- Servers are categorized by stratum
  - Stratum 0 knows exact time
    - Atomic clocks like NIST and GPS
  - Stratum 1 slaves time to Stratum 0
    - Best is GPS/GAL, others WWV/DCF/JJY, etc
  - Stratum 2 exchanges network time to Stratum 1
    - millisecond accuracy

Questions?