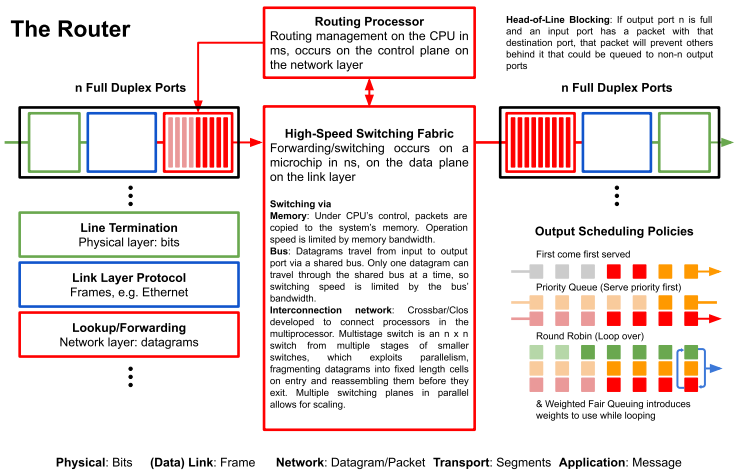

Obtaining an IP Using DHCP: A Play **DHCP Discover** Client: Is there a DHCP server out there? *Source IP* (no IP assigned): 0.0.0.0, *Destination IP* (broadcast): 255.255.255.255 **DHCP Offer** DHCP Server: Yes! I'm a DHCP server. Here's an IP address you can use: 170.188.2.154 *Source IP*: [Router's IP], *Destination IP* (broadcast): 255.255.255.255 **DHCP Request** Client: Great! I'll be using that *Source IP* (no IP assigned): 0.0.0.0, *Destination IP* (broadcast): 255.255.255.255 **DHCP ACK** DHCP Server: Sweet, it's all yours. *Source IP*: [Router's IP], *Destination IP* (broadcast): 255.255.255.255
Address Resolution Protocol, A Play: Device A wants to send Device B a datagram. A checks its ARP table, which does not contain B's information. A uses ARP to find B's MAC address by broadcasting a query packet containing B's IP address. This query is received by all nodes on the LAN. B contains a node in the LAN, receives the query, and replies to it with its MAC address. A receives its reply, adding B as an entry to its ARP Table. A can now send Device B the datagram using the information stored on its ARP table.
Transport Layer Security (Given sender S and recipient R): S wants to establish a secure TCP connection with R, verify that R is really R, send R a master secret key which will be used by both R and S to generate all the symmetric keys they need for the TLS session.
Almost-TLS Initial Handshake: A Play. S: TCP SYN (TCP connection, no payload). R: TCP/SYNACK (TCP connection, no payload). S: TCP ACK (TCP connection, may contain payload). S: TLS hello. R: Certificate (proof that R is R). S: EMS = K_{auth}+(MS).

IPv4: 20-60 byte header (depending on options: timestamp, record route taken) vs **IPv6**: No fragmentation/reassembly at intermediate routers (only at source and destination). No checksum is natively supported, so error checking must be performed by hardware and protocols. No options (available as upper-layer, next-header protocol at router). IPv4 datagrams can carry IPv6 datagrams in their payload (tunneling).

Subnets: Device interfaces that can physically reach each other without passing through an intervening router. **Classless InterDomain Routing** (CIDR): Devices in a subnet have a LAN address: a.b.c.d/(x bits for subnet identifier)

Network Address Translation (NAT): All devices on the same network have the same external IP address (NAT IP address). Outgoing datagrams have their source IP address and port number replaced with a NAT IP address and new source port that are mapped in a NAT translation table that the router uses to replace the IPs and ports of incoming and outgoing packets. Incoming datagrams have their IP and port replaced back to the original.

Link State message complexity: N routers, O(n2) messages sent. Speed of convergence may have oscillations. Router can advertise incorrect link cost. Each router only computes its own table. Open Shortest Path First (OSPF).

Distance Vector message complexity: Exchange between neighbors. Speed of convergence varies. Router can advertise incorrect path cost to neighbors and cause black-holing. Each router's DV is used by other routers, as information is exchanged between neighbors, errors will propagate through the network.

Autonomous Systems is an internet approach to scalable routing where routers are aggregated into regions known as "autonomous systems" (AS) (a.k.a. "domains"); with each AS consisting of a group of routers that are under the same administrative control. The routing algorithm running with an autonomous system is called an intra-autonomous system routing algorithm. The most common intra-AS routing protocols are: Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), and OSPF.

Authentication: In OSPF Routing, with Authentication, only trusted routers can participate in the OSPF protocol within the autonomous system; hence, un-authorized devices will not be able to interject their potentially malicious entries into router tables.

Border Gateway Protocol (BGP) "glues" thousands of autonomous systems (technically ISPs) together. Pairs of routers exchange routing information over TCP connections over port 179, and is decentralized, asynchronous, and in the vein of DV routing. BGP routes using CIDRized destination addresses. Each Autonomous System has gateway routers and internal routers. Gateway routers are on the edge of the AS that directly connects to one or more routers on the inside while an internal router only connects to the host and routers within its AS. Sessions are established via a semi-permanent TCP connection. Sessions spanning two Autonomous Systems are called an external BGP (eBGP). Ones spanning routers in the same AS are called internal BGP (iBGP).

Hot Potato Routing Lifecycle: **1.** Learn from inter-AS protocol that subnet x is reachable via multiple gateways. **2.** Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways. **3.** Choose a local gateway that has lease cost to the NEXT-HOP router. **4.** Determine from the forwarding table the interface l that leads to least-cost gateway. Enter (x, l) in forwarding table.

Internet Control Message Protocol (ICMP): Used by hosts and routers to communicate network-level information. Error reporting (i.e. unreachable host/network/port/protocol), echo request/reply. Network layer "above" IP: ICMP messages carried inside IP datagrams' payload. Message format: type + code + first 8 bytes of IP datagram causing error. Datagrams + UDP segments can determine the names/addresses of the routers and switches between the source and destination.

Traceroute sends sets of UDP segments to the destination. When the ICMP message arrives at the source, the round trip times are recorded

Switches mainly operate at the Link Layer. They use hardware, microchips (ns). Frames are exchanged. Majority do not read network-layer datagrams. No access to the network-layer addresses, do not use routing, they use OSPF. Link Layer addresses are used to forward frames using MAC addresses. They populate their forwarding table using flooding and MAC addresses. **Routers** mainly operate at the Network Layer. They use software, CPUs (ms). They read network-layer datagrams, examine link-layer headers, and use routing algorithms like LS and DV. **Both** are store and forward (examine network layer headers) and have forwarding tables.

Software Defined Networking SDN control plane routing functionality is separated from the physical router, as it only performs forwarding. Remote controller computes and distributes forwarding tables. This is implemented in software.

(Global) Centralized Routing Algorithms compute the least-cost path between a source and destination using complete, global knowledge about the network. That is, the algorithm takes the connectivity between all nodes and all link costs as inputs. This then requires that the algorithm somehow obtain this information before actually performing the calculation. The calculation itself can be run at one site or could be replicated in the routing component of each and every router. (LS)

Decentralized Routing Algorithms calculate of the least-cost path is carried out in an iterative, distributed manner by the routers. No node has complete information about the costs of all network links. Instead, each node begins with only the knowledge of the costs of its own directly attached links. (DV)

Simple Network Management Protocol is an Application Layer protocol used to convey network-management control and information messages between a managing server and an agent executing on behalf of a managing server. Requests typically are used to query or modify Management Information Base (MIB) object values associated with a managed device. Agents also can send unsolicited "trap" messages to the managing server to notify the server of exceptional situations.

Link Layer Services: **Framing** enables datagram encapsulation into a frame prior to sending it over the link. **Link access** uses MAC addresses to specify source and destination. This is not the same as IP addresses, which are used at the network layer. **Reliable delivery**; sometimes link layer protocols guarantee that datagrams will be delivered without error. However, reliable delivery is not usually used in the link layer if the link's error rate is low anyway.

For example, the link does not ensure reliable delivery if fiber optic cables are used, since fiber optic is not prone to electromagnetic interference anyway.

Network Interface Card The link layer is most often implemented in hardware by the NIC (network interface card). The NIC is attached to a computer via its bus. The computer sends datagrams to the NIC, which automatically encapsulates the datagrams into frames and transmits them to the next link. When the frame is being sent, error checking and flow control bits are also set. When the NIC receives a frame, it extracts the datagram contained in the frame, performs error detection, and passes the datagram to the computer.

Error Detection: At the link layer, the data is augmented with EDC bits (error correction and detection). At the link layer, error detection and correction is performed in hardware rather than software. Remember that error detection is not 100% reliable at the link layer, which is why network-layer protocols like UDP and TCP also perform error detection and/or correction. Error detection can be accomplished using parity bits or CRC. Single bit parity is good for detecting a single flipped bit, and two-dimensional parity is able to both detect and correct single bit errors.

IP Address: 32-bit, network-layer address for interface used for forwarding "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense). **Media Control Access** (MAC) addresses: 48-bit (for most LANs) permanent, hard-coded addresses in the NIC, also sometimes software settable, unique, and administered by IEEE. Manufacturers buy a range of MAC addresses to assign to individual devices.

Address Resolution Protocol (ARP): Each IP node in hosts and routers on a LAN has a table with IP/MAC address mappings for some of the LAN nodes formatted < IP address; MAC address; TTL >, which includes the mapping's time to live (after which the mapping will be forgotten; the times typically begin at 20 minutes).

Domain Name Servers resolve host names for hosts anywhere on the internet while ARTP only resolves IP addresses for hosts and routers on the same subnet that the query is made.

Ethernet Frame Structure: *Preamble*: Wakes up the receiving device and syncs the receiver's and sender's clocks, *Dest. Address*: The destination MAC address of the receiver, *Source Address*: The destination MAC address of the sender, *Type*: Used to multiplex ethernet, *Payload*: Data, *Cyclic Redundancy Check* for error detection appends a pattern at the end of the frame. Bits have flipped if the $m\%2 \neq 0$.

Virtual Local Area Networks (VLANs) can be used to either isolate devices connected to a single switch or to connect multiple switches together, which allows them to be used like a single switch. *Switch supports VLANs*: multiple virtual LANs can be defined over a single physical LAN infrastructure. Hosts then communicate with each other as if they were connected to the switch. Port-based VLANs switch ports divided into groups by the network manager. *Three drawbacks*: lack of traffic isolation, inefficient use of switches, and complexity managing users

CIAA: **Confidentiality**: Only send, intended receiver should "understand" message contents. Sender encrypts, receiver decrypts message. **Integrity**: Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection; recall checksums and hashing. **Authentication**: Sender, receiver want to confirm identity of each other (digital signatures). **Access/Availability**: Services must be accessible and available to users.

Hash Properties: Many-to-one, produces fixed-size message digest (fingerprint), impossible un-hash. **Most used**: *MD5/Ron Rivest* is in use today; 128-bit hash using four step process with a padding step (1+n zeroes), then the appending of the 64-bit representation before the padding, accumulator, and then 4 rounds of mangling of 16-word blocks. *SHA-1*, similar to MD4, predecessor of MD5; US federal standard, 160-bit message digest.

Authentication key & Message Authentication Code (MAC): For message integrity, in addition to hashing, the sender and recipient need a shared key S. This is a string of bits that comprise an authentication key. 1. The sender creates a message m, contacts S with m (m+S), and hashes it H(m+S). This is the Message Authentication Code (MAC). 2. The sender then appends the MAC to the message (m, H(m+S)) and sends it to the recipient. 3. The recipient receives the extended message (m, h), and can calculate the MAC as it knows S to compare the hashes

Digital Signatures: To authenticate messages, digital signatures can be used. A sender can encrypt a message m with their private key Kpr, which will produce a signature. This signature can then be sent over with the message, which allows the recipient to use the sender's public key to decrypt the signature and compare with the original message. **Note**: It's expensive to sign messages via encryption/decryption, so it's preferred to hash the original message before signing it. This will generate a fixed-length, more efficient "fingerprint". The sender signs the hash of the message as opposed to the message itself, as the hash output is much smaller than the original message, making the signature easier to compute.

The Public Key Certification Authority binds public keys to a particular entity to verify that entities are who they are claiming to be.

Securing Email (Pretty Good Privacy): Given a sender S, recipient R, message m, hash function H, and sender's private and public keys Kpr and Kpub: S wants to securely email m to R. S will H(m) to obtain the message digest d. S then signs H(m) with Kpr to obtain a digital signature Sd, they then create mnew=m+Sd, and send it to R.

IPSec & VPN & Protocols: **Authentication Header** (AH) protocol provides source authentication and data integrity but not confidentiality. **Encapsulation Security Protocol** (ESP) provides source authentication, data integrity, and confidentiality. It is more widely used than AH, as confidentiality is a huge part of VPNs.

Firewalls isolate organization's internal network from larger Internet, allowing some packets to pass, blocking others. Useful in preventing denial service attacks, illegal modification/access of internal data, and only allowing authorized access inside network. *Types*: **Stateless** packet filters (Internal network connected to the internet via router firewall. Filters packet-by-packet, decision to forward / drop packet based on source IP address, destination IP address, TCP / UDP source, destination port numbers, ICMP message type, TCP SYN, ACK bits), and **Stateful** packet filters (Track status of every TCP connection. Determines whether incoming, outgoing packets "make sense" by tracking connection setup (SYN), teardown (FIN). Timeouts inactive connections at firewall so that it no longer admits packets) , and **Application Gateways** (Filter packets on application data as well as on IP/TCP/UDP fields). Main limitations: IP Spoofing (Router can't know if data "really" comes from claimed source), if multiple apps need special treatment, each has own app, client software must know how to contact gateway (e.g. must set IP address of proxy in Web browser). Filters often use all or nothing policy for UDP. Tradeoff: Degree of communication with outside world, level of security Many highly protected sites still suffer from attacks.

Intrusion Detection and Prevention Systems (IDPS): Deep Packet Inspection: Look at packet contents (e.g., check character strings in packet against database of known virus, attack strings) Examine correlation among multiple packets. Port scanning, network mapping, DoS attack, Packet filtering: Operates on TCP/IP headers only. No correlation check among sessions.

Web Request: A device wants to connect to google.com. First, it needs to acquire its own and the router's IP address using DHCP. The DHCP request is encapsulated in a UDP datagram, encapsulated in an IP datagram, encapsulated in an Ethernet frame. Since the DHCP request broadcasts to 255.255.255.255, the Ethernet frame's destination MAC address is set to FF:FF:FF:FF:FF:FF, which is then received by the router running a DHCP server. When the router receives the Ethernet frame, the IP datagram is extracted from the frame, and the UDP datagram is extracted from the IP datagram. The DHCP request is finally extracted from the UDP datagram. Then, the router creates a DHCP ACK, which is encapsulated in a UDP datagram, encapsulated in an IP datagram, encapsulated in an Ethernet frame, which is sent to the broadcast address and received by the client. Now, the client has its own IP address, the IP address of the DCHP router, and the DNS server(s). Now, the client needs to get the IP address of google.com, so it sends a DNS query to a DNS server. The query is encapsulated in a UDP datagram, encapsulated in an IP datagram, encapsulated in an Ethernet frame. However, the client cannot send the frame to the LAN router without knowing its MAC address. To discover the router's MAC address, the client sends an ARP query to the broadcast address. Then, the LAN router replies with an ARP reply containing its MAC address. Now that the client knows the LAN router's MAC address, it can finally send the DNS request to the LAN router to be routed. The client sends the packet containing the DNS query to the router. The router then forwards the IP datagram to the ISP's network. The datagram is then routed through the ISP's network using RIP, OSPF, IS-IS, and BGP. Finally, the DNS server receives the DNS query and replies to the client with the IP address of google.com. Now that the client has the IP address of google.com, it can send an HTTP request to Google at last. The client then opens a TCP socket with the IP obtained via DNS and port 80 (reserved for HTTP traffic). The TCP connection is established with the three-way handshake, starting with an SYN segment sent from the client to Google's servers. Google's servers then reply to the SYN with an SYN ACK. Then, the client finishes the handshake with an ACK segment. Now that the TCP connection is established, the client sends an HTTP request to Google. When Google's servers receive the request, they reply with an HTTP response containing an HTML file.