THOMAS MORE | UNIVERSITY OF APPLIED SCIENCES

THOMAS MORE | University of Applied Sciences

Computing Fundamentals

Planning

Session	Subject	Test – Hand-in
1	Network Models	
2	Internet Protocol Suite	
3	Network segmentation	
4	Network protocols	
5	Operating systems	
6	Command Line	
-- 30/10 – 5/11 --	Autumn break – HERFSTVAKANTIE	
7	Command Line	
8	Mid-term test	**Test**
9	**Scripting**	
10	Virtualization - Cloud computing	

Network Protocols

- **What is a network protocol?**
- A network protocol is a set of rules that govern data communication between different devices in the network. It determines what is being communicated, how it is being communicated, and when it is being communicated. It permits connected devices to communicate with each other, irrespective of internal and structural differences.

How do Network Protocols Work?

- It is essential to understand how devices communicate over a network by recognizing network protocols. The **Open Systems Interconnection** (**OSI**), the most widely used model, illustrates how computer systems interact with one another over a network. The communication mechanism between two network devices is shown by seven different layers in the OSI model. Every layer in the OSI model works based on different network protocols. At every layer, one or more protocols are there for network communication. To enable network-to-network connections, the **Internet Protocol** (**IP**), for example, routes data by controlling information like the source and destination addresses of data packets. It is known as a network layer protocol.

Main types of network protocols

- Network protocols serve different primary purposes, which allow us to categorize them into main functional types. It’s helpful to understand the role of each type within the bigger picture.

Main types of network protocols

- **Communication protocols**
  Like a postal service delivering letters and packages between homes, communication protocols enable devices to exchange messages and data payloads across networks. They establish rules and conventions for reliable data transfer from senders to recipients.
  Examples include **TCP/IP**, the core delivery protocol of the Internet, **FTP** for **file transfers**, and **SMTP** for **email**.
  Communication protocols enable vital networked applications and services we use daily, such as websites, email, file sharing, media streaming, and more. They form the basic transport mechanisms for connectivity.

Main types of network protocols

- **Network security protocols**
  If communication protocols are the messengers moving data between devices, network security protocols act like security guards regulating access and protecting the data flows.
  Security protocols employ measures like encryption and authentication to secure communications from eavesdropping or harmful tampering, much like security personnel safeguard facilities.
  Protocols like **SSH**, **SSL/TLS**, and **IPsec** create secure tunnels to shield data and validate identities. Just as guards check badges at facility gates, security protocols block unauthorized access.

Main types of network protocols

- **Network management protocols**
  Network management protocols enable administration capabilities to monitor performance and remotely configure network equipment.
  They're like IT administrators overseeing the health of devices across networks and managing configurations. Protocols like **SNMP** and **ICMP** provide insights into network status to diagnose and troubleshoot issues.
  Others like **DHCP** and **DNS** dynamically assign IP addresses and map device names.
  Management protocols grant networks greater robustness and control just as capable administrators keep enterprise IT infrastructure working.

Communication protocols

- **1. Transmission control protocol/Internet protocol (TCP/IP)**
  The Transmission Control Protocol/Internet Protocol (TCP/IP) suite facilitates communication by handling the encapsulation of data into packets at the sender, transmitting it reliably across networks, routing packets to the destination address via intermediary network devices like routers and switches using Internet Protocol (IP), before finally reassembling packets in the proper order at the receiving host.
  Built-in error-checking capabilities automatically request the retransmission of any missing or corrupted packets. This provides reliable end-to-end connectivity crucial for virtually all modern applications to function well over LANs and the Internet alike.
  However, attacks such as SYN floods and IP spoofing may attempt to undermine availability by overwhelming target systems with bogus requests or impersonating trusted devices.

Communication protocols

- **2. User datagram protocol (UDP)**
  User Datagram Protocol (UDP) offers a lean alternative to TCP. It essentially trades reliability for speed, proving useful for time-sensitive purposes like video calling, streaming media, and online gaming. By skipping error-checking steps, UDP saves processing overhead and accelerates data transfers, delivering packets quickly but without guarantees.
  Thus, packets may arrive out-of-order or go missing altogether. While acceptable for streaming audio/video applications able to tolerate such losses, UDP can be risky in transferring critical data. Security mechanisms are also lacking, making encrypted alternatives more secure for sensitive applications.

Communication protocols

- **3. File transfer protocol (FTP)**
  The venerable File Transfer Protocol (**FTP**) still sees widespread use for uploading and downloading files between client and server over a TCP/IP network. Web hosting environments often employ **FTP** to provide multiple contributors with file access for routine website updates.
  FTP options like directory listings and non-interactive transfers make batch operations convenient. However, privacy is sorely lacking as login credentials and data transfer in cleartext without encryption. The encrypted **FTPS** variant addresses this issue for use cases requiring tighter security.

Communication protocols

- **4. Session initiation protocol (SIP)**
  SIP serves as the backbone for multimedia sessions over IP networks. So, while analog signals ran over telephone networks before, SIP helped "packetize" voice and media into data that can be transmitted digitally. From modern video conferencing apps to entire cloud phone systems used by large contact centers, **SIP** makes all of that possible. It handles the nitty-gritty signaling, session management, and teardown details so audio, video, and other media exchange happen smoothly across the internet. SIP integrates with standardized voice and video protocols to make internet-based real-time communications possible.

Network security protocols

- **5. Secure shell (SSH)**
  Secure shell, commonly known as **SSH**, is one of the most prevalent network protocols used today. It enables secure remote login connections to devices like servers, switches, and firewall appliances from client software.
  SSH sets up an encrypted tunnel to protect the authentication session and subsequent remote access from eavesdroppers. It prevents plainly transmitting credentials that could be intercepted. **SSH** replaces older insecure protocols like Telnet and rlogin, which are still sometimes used but lack encryption.
  Over the years, **SSH** has become a Swiss-army knife network tool that administrators worldwide rely on for tasks like securely transferring files with **SFTP** and tunneling or port forwarding network traffic. Commercial SSH implementations boast advanced features out of necessity focused on availability, compliance, and threat prevention. For example, granular access controls, detailed session logging, and host key management integrate SSH deeply with identity and authentication ecosystems while responding to vulnerabilities.

Communication protocols

- **6. Secure sockets layer (SSL) / Transport layer security (TLS)**
  Transport layer security (**TLS**) and its older relative, the **Secure sockets layer** (**SSL**), implement cryptographic protections for data in transition across networks. Encrypting application traffic and authenticating connecting parties using certificates prevents tampering and eavesdropping.
  Numerous applications transparently overlay **TLS/SSL** without users noticing. Examples include **HTTPS** web browsing, **Secure shell** encrypted terminal sessions (SSH), **Virtual Private Network** tunnels (VPN), and **secure email**.
  However, TLS is **vulnerable** because, when faced with certain attacks, it devolves connections to a **weaker encryption**. Keeping software up-to-date and properly **validating certificate chains** maintains robust security foundations for applications.

Communication protocols

- **7. Secure FTP (FTPS)**
  FTP enjoys longevity, securing crucial yet predictable file transfer needs globally across industries. **FTPS** supercharges **FTP** by adding **SSL/TLS**-based encryption, mitigating data theft or tampering risks over unprotected transfers. Supporting the latest cryptographic agility and enforcement practices is key, given lengthening **data retention** mandates. With growing runtime exploitation and malware threats, however, content scanning before final storage brings vital assurances. Overall, while **FTPS** brings necessary encryption, centralized access controls and integration with funnel points like proxies and firewalls ensure governance beyond just the wire.

Communication protocols

- **8. Email protocols**
  POP stands for Post Office Protocol. The POP protocol was published in the year 1984. POP has been updated two times namely "**POP2**" and "**POP3**". The POP protocol is an Internet Standard Protocol that works on the application layer. It is used to get an access email from the mail server. The need for POP mainly arises when the user or client does not have a continuous internet connection and what's to receive email messages. The Pop client makes use of POP to pull email messages from the POP server. POP3 is the updated version of POP. **SMTP**, **IMAP**, **POP**, and **POP3** are some of the email protocols. Each type of protocol used has a specific mechanism.

Network management protocols

- **9. Simple network management protocol (SNMP)**
  Simple network management protocol or **SNMP** is widely used for monitoring and managing all sorts of network-connected devices—from routers and switches to printers, firewalls, and servers. It works by letting an SNMP manager send queries to devices being monitored, which each have a small piece of SNMP agent software installed to collect status and performance data. SNMP can track valuable telemetry like uptime stats, link utilization, errors spotted, and plenty more. The agents gather all this and report back to monitoring tools so network administrators can get a nice centralized view instead of checking individually. SNMP even supports alerts and notifications for faults or thresholds being crossed, known as SNMP traps, namely messages sent from monitored devices to the manager to indicate an anomaly or issue. When it comes to SNMP, there are a few different versions available. SNMPv2 and SNMPv3 are the most common currently in use. **SNMPv3** is more advanced, introducing stronger security with encryption and authentication, unlike the older SNMPv2.

Network management protocols

- **10. Internet control message protocol (ICMP)**
  The **Internet Control Message Protocol** (**ICMP**) handles basic diagnostic functions like querying whether destinations are reachable and responding with status updates. **ICMP** is best recognized as enabling ubiquitous "**ping**" connectivity verification requests emitted by network troubleshooting tools, triggering target devices to **report back timing and availability data**.
  By default, networks usually permit **ICMP** packets as blocking will obstruct vital network monitoring. However, **excess ICMP** traffic can sometimes be exploited to flood networks in **denial-of-service** brinksmanship. Like **UDP**, **ICMP eschews hard security**, given it predates modern encryption.

Network management protocols

- **11. Address resolution protocol (ARP)**
  Devices communicate via **MAC addresses** on local networks. But we often only know IP addresses for destinations. **The address resolution protocol** bridges this gap by **resolving IP addresses to associated MAC addresses** that network adapters use. Your computer maintains an ARP table caching these mapped addresses locally.
  When you try reaching a new destination, broadcasts seek the MAC for the IP, update the table after getting a reply, and subsequently transmit. Under the hood, the process is invisible to users. Unfortunately, ARP lacks authentication natively, meaning cache entries can get overwritten by spoofing attacks. Protecting ARP behavior is thus important for robust connectivity on local network segments.

Network management protocols

- **12. NetFlow**
  Where SNMP focuses on device-level statistics, **NetFlow** lets you step up to network-wide monitoring and analysis by understanding traffic patterns coursing through your infrastructure. By processing flow-based data about connections, the volume, timing, directionality, duration, endpoints, and applications involved all become transparent.
  Network flow records cement visibility into network usage and dependency, empowering informed decisions about capacity, security, layout, and more.
  The catch lies in NetFlow compliance—next-gen firewalls and web proxies generally work, but switches and routers need capabilities enabled. Otherwise, blind spots manifest where critical flows lack visibility.

Network management protocols

- **13. sFlow**
  sFlow is a packet sampling technology used for monitoring network devices like routers, switches, and wireless access points across vendors. Unlike NetFlow, which samples full packet flows, sFlow randomly samples 1 out of N individual packets passing through an interface. This sampling occurs at wire speed via dedicated hardware chips embedded in the network devices.
  A sFlow software agent combines the sampled packet data with interface counters and forwards table info into sFlow datagrams. These datagrams are shipped off to a central sFlow collector for analysis. So, while less comprehensive than NetFlow, sFlow provides network-wide visibility with quantifiable accuracy, especially for bandwidth-heavy traffic like streaming video.

Network management protocols

- **14. Border gateway protocol (BGP)**
  On the wild internet, **BGP** helps tame things by managing routing data exchange between organizations. It essentially maintains a large-scale map of network reachability between autonomous systems that agree to share access. Your ISP likely participates, enabling your traffic to traverse multiple networks when accessing websites abroad.
  By distributing routing updates, BGP-enabled routers know which paths packets should traverse to reach intended destinations. Validating route announcements and **preventing malicious hijacking** is thus crucial for BGP security and reliability mechanisms. Within corporate networks as well, BGP is key for connecting privately managed subnets and sharing route data.

Network management protocols

- **15. Domain name system (DNS)**
  DNS functions as a phonebook for the Internet. This protocol translates domain names that humans can easily remember, like google.com or wikipedia.org, into numerical IP addresses that computers and routers use to fetch the correct websites and content. It essentially matches names with the right numbers.
  A breakdown can occur if the DNS information that maps a domain to an IP gets somehow modified or corrupted by an attack. When the DNS “records” providing this name-to-address mapping get poisoned or altered, browsers and apps can get misdirected. This unfortunately remains one of the simpler ways even massive websites go offline suddenly.

Network management protocols

- **17. Dynamic host configuration protocol (DHCP)**
  Networks keep functioning smoothly in part thanks to DHCP performing helpful background work across all those routers, switches, servers, and devices. Whenever endpoints like laptops, phones, or tablets connect to the wireless network or plug into the wired LAN ports in office buildings, DHCP does the housekeeping work of assigning them valid IP addresses plus other critical networking information as part of the join process.
  By automatically allocating IP addresses instead of manual configuration, DHCP simplifies management for devices ranging from home routers to complex enterprise networks. However, if something goes wrong, such as a rogue DHCP server appearing on the network and interfering or a denial-of-service attack flooding a DHCP server with fake requests, new devices won’t be able to obtain the settings they need to get online.

Network management protocols

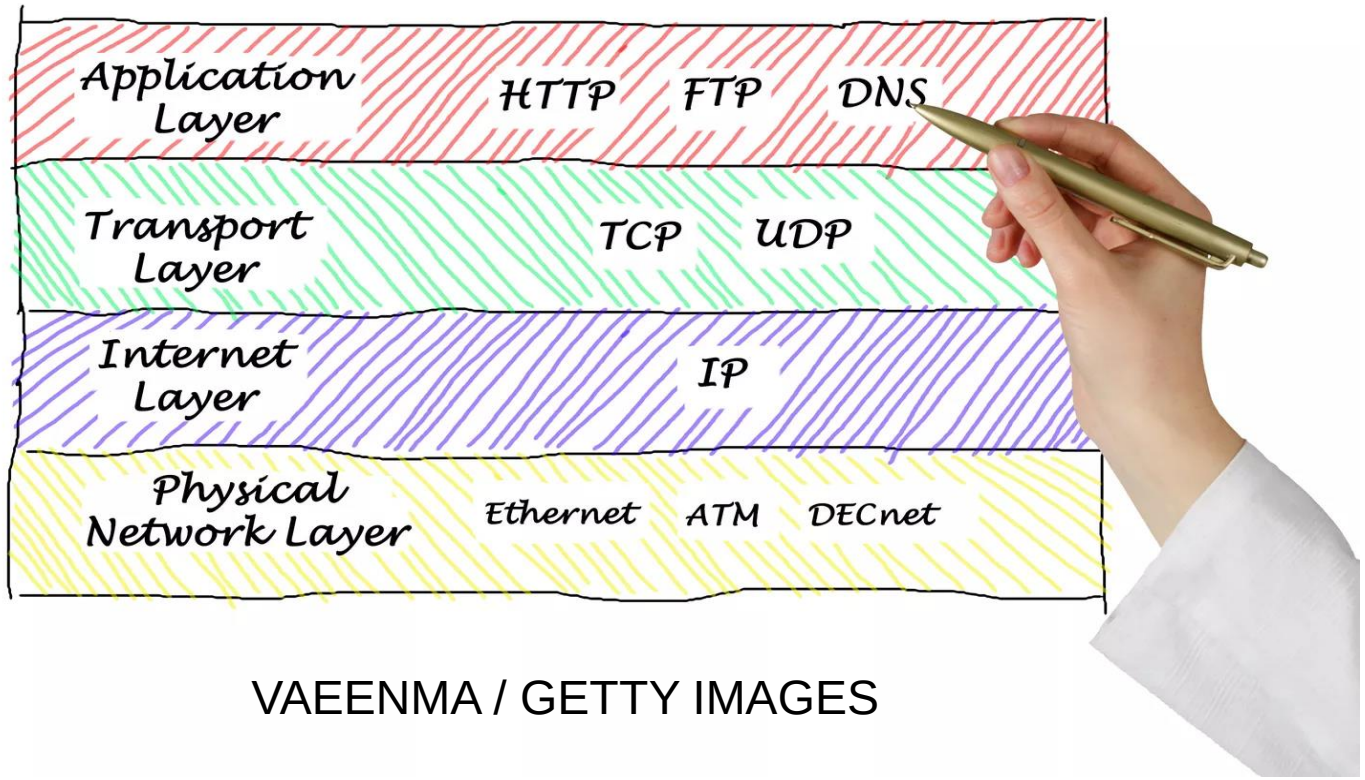
- **18. Telnet**
  Despite security shortcomings compared to newer tools, **Telnet** remains a useful way to remotely access devices over the network. It enables commands to be run on routers, switches, or servers from another system with network connectivity as though seated at the target device's console, which is handy for tasks like tweaking settings or grabbing debug logs.
  However, everything gets exchanged between the admin's PC and the managed device in plain, unencrypted text, including account credentials and privileged commands. **Sniffing the traffic** makes stealing sensitive information extremely easy, meaning Telnet access requires additional controls like **jumpboxes** to limit exposure.

Common Ports

- The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) each use port numbers for their communication channels. The ports numbered 0 through 1023 are the well-known system ports, reserved for special uses. **Port 0** is not used for TCP/UDP communication although it used as a network programming construct.

Computing Fundamentals

Common Ports



VAEENMA / GETTY IMAGES

Computing Fundamentals

Well-Known Port Numbers

Service, Protocol, or Application	Port Number	TCP or UDP
FTP (File Transfer Protocol)	20, 21	TCP
SSH (Secure Shell Protocol)	22	TCP
Telnet	23	TCP
SMTP (Simple Mail Transfer Protocol)	25	TCP
DNS (Domain Name System	53	UDP
TFTP	69	UDP
HTTP	80	TCP
POP3	110	TCP
IMAP4	143	TCP
HTTPS	443	TCP