UNIVERZITA J. E. PURKYNĚ V ÚSTÍ NAD LABEM
Přírodovědecká fakulta

Internetové technologie a protokoly

KI/ITP

RNDr. Jan Krejčí, Ph.D.

Ústí nad Labem 2020

Kurz:	Internetové technologie a protokoly
Obor:	Informační systémy, Informatika (dvouoborové studium).
Klíčová slova:	model ISO/OSI, model TCP/IP, síťový protokol, konzole
Anotace:	Kurz je zaměřen na aplikační vrstvu modelu TCP/IP a jí odpovídajícím vrstvám referenčního modelu ISO/OSI. Student se v rámci přednášky seznámí s nejčastěji využívanými protokoly a službami na síti Internet. V rámci praktických seminářů bude student konfigurovat jednotlivé služby a sledovat počítačovou komunikaci, ve které bude detekovat a analyzovat jednotlivé protokoly a jejich pakety.

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

© Katedra informatiky, PřF, UJEP v Ústí nad Labem, 2020

Autor: RNDr. Jan Krejčí, Ph.D.

Obsah

Úvodní slovo

Kurz je zaměřen na aplikační vrstvu modelu TCP/IP a jí odpovídajícím vrstvám referenčního modelu ISO/OSI. Student se v rámci přednášky seznámí s nejčastěji využívanými protokoly a službami na síti Internet. V rámci praktických seminářů bude student konfigurovat jednotlivé služby a sledovat počítačovou komunikaci, ve které bude detekovat a analyzovat jednotlivé protokoly a jejich pakety.

- Písemný test se skládá z otázek odpovídajících sylabu kurzu

bodové hodnocení	známka
100 % – 91 %	1
90 % – 81%	2
80 % – 71%	3
70 % – 0%	4

- Ústní zkouška:
  - student losuje 2 otázky
  - min. 15 minut samostatné přípravy
  - odpověď na každou z otázek je hodnocena známkou 1 – 4
- Výsledná známka je celkovým zhodnocením všech tří známek.
- Zkouška je považována za neúspěšnou, pokud je student hodnocen stupněm 4 z jakékoli části.

Součástí opory je přiložen studijní materiál Ing. Růžičky (příloha A), který je určen studentům středních škol se zaměřením na IT. Studenti předmětu Internetové technologie a protokoly, který potřebný středoškolský základ nemají, by si materiál měli prostudovat v rámci přípravy na tento předmět.

Příklad písemné části zkoušky

1. Jakou úlohu plní fyzická vrstva v referenčním modelu OSI? (5)
2. Jakou úlohu plní linková vrstva v referenčním modelu OSI? (5)
3. Jakou úlohu plní síťová vrstva v referenčním modelu OSI? (5)
4. Jakou úlohu plní transportní vrstva v referenčním modelu OSI. (5)
5. Vyjmenujte a popište jaké služby poskytuje aplikační vrstva (alespoň 4). (5)
6. Popište a vysvětlete jak funguje TCP protokol. (5)
7. Popište a vysvětlete jak funguje UDP protokol. (5)
8. Popište jak funguje Secure Sockets Layer. (5)
9. Co je RDP? (5)
10. Jaký je rozdíl mezi jednotlivými protokoly a jaké mají výhody či nevýhody? (5)
11. Popište Telnet. (5)
12. Popište protokol IMAP. (5)
13. Popište protokol POP. (5)

1 Opakování: ISO/OSI, TCP/IP

CÍLE KAPITOLY

V této kapitole si zopakuje referenční model ISO/OSI a TCP/IP.

KLÍČOVÁ SLOVA

OSI model, fyzická vrstva, linková vrstva, síťová vrstva, transportní vrstva, relační vrstva,prezentační vrstva, aplikační vrstva, TCP/IP

1.1 ISO/OSI

Náplň této kapitoly je obsažen v Úvod do počítačových sítí I.[1, strana 15-19] kapitola 2.4. Vrstový model ISO/OSI a v Počítačové sítě I. [2, strana 61-105] kapitola 3 Referenční model OSI.

1.2 TCP/IP

Náplň této kapitoly je obsažen v Úvod do počítačových sítí I.[1, strana 20-21] kapitola 2.6 Vrstvový model TCP/IP a v Počítačové sítě I. [2, strana 107-135] kapitola 4 TCP/IP Model.

SHRNUTÍ

Zopakování látky probrané v ZPP.

OTÁZKY

1. Jakou úlohu plní fyzická vrstva v referenčním modelu OSI?
2. Jakou úlohu plní linková vrstva v referenčním modelu OSI?
3. Jakou úlohu plní síťová vrstva v referenčním modelu OSI?
4. Jakou úlohu plní transportní vrstva v referenčním modelu OSI?
5. Jakou úlohu plní relační vrstva v referenčním modelu OSI?
6. Jakou úlohu plní prezentační vrstva v referenčním modelu OSI?

7. Jakou úlohu plní aplikační vrstva v referenčním modelu OSI?
8. Jakými dvěma typy komunikace je možné zasílat segmenty?
9. Vysvětlete výhody jednotného modelu pro komunikaci v počítačové síti.
10. Na jaké vrstvě OSI modelu hovoříme o paketu?
11. Na jaké vrstvě OSI modelu hovoříme o rámci?
12. Na jaké vrstvě OSI modelu hovoříme o segmentu?
13. Jakou úlohu plní vrstva síťového přístupu v TCP/IP modelu?
14. Jakou úlohu plní vrstva internet v TCP/IP modelu?
15. Jaký je rozdíl v protokolech TCP a UDP?
16. Jaké znáte významné protokoly v aplikační vrstvě?

MÍSTO PRO VAŠE POZNÁMKY

2 Aplikační vrstva

CÍLE KAPITOLY

Opakování a bližší seznámení s aplikační vrstvou, popis a její funkce.

KLÍČOVÁ SLOVA

aplikační vrstva

Náplň této kapitoly je obsažen v Úvod do počítačových sítí I.[1, strana 17] kapitola 2.4.7. Aplikační vrstva (Application Layer) a v Počítačové sítě I. [2, strana 104, 162-135] kapitoly 3.7 Aplikační vrstva a 4.4 Aplikační vrstva.

SHRNUTÍ

Aplikační vrstva Obecně zajišťuje v TCP/IP modelu širokou množinu služeb, které na jedné straně přímo komunikují s uživatelem počítačového systému a an druhé straně s nižší transportní vrstvou. Ze širokého portfolia aplikačních protokolů implementovaných do modelu TCP/IP jsme si popsali jen ty nejzákladnější, na které spoléhá provoz lokálních počítačových sítí a Internetu již desítky let. Tato množina se však trvale rozrůstá na základě nových požadavků uživatelů a jejich implementací vývojovými pracovníky.

OTÁZKY

1. Jakou úlohu plní aplikační vrstva v referenčním modelu OSI?
2. Jakou úlohu plní aplikační vrstva modelu TCP/IP?
3. Vyjmenujte a popište jaké služby poskytuje aplikační vrstva. (alespoň 4)

MÍSTO PRO VAŠE POZNÁMKY

3 Protokoly Transportní vrstvy ISO/OSI

CÍLE KAPITOLY

Opakování a bližší seznámení se s protokoly transportní vrstvy TCP a UDP.

KLÍČOVÁ SLOVA

transportní vrstva, TCP, UDP

Náplň této kapitoly je obsažena v Počítačové sítě I. [2, strana 116] kapitoly 4.3.3 Protokol TCP a 4.3.4 Protokol UDP. Dále je pak protokol TCP zpracován v A.15 a protokol UDP v A.16.

SHRNUTÍ

Zopakování látky probrané v ZPP.

OTÁZKY

1. Popište paket odeslaný pomocí TCP protokolu.
2. Popište paket odeslaný pomocí UDP protokolu.
3. Popište a vysvětlete jak funguje TCP protokol.
4. Popište a vysvětlete jak funguje UDP protokol.
5. Jak funguje v TCP protokolu navázání spojení?
6. Porovnejte TCP a UDP protokol a vysvětlete, kde a proč se jaký protokol používá.

MÍSTO PRO VAŠE POZNÁMKY

4 Protokoly Relační vrstvy ISO/OSI

CÍLE KAPITOLY

Seznámení se s protokoly PAP a SSL pracující na relační vrstvě.

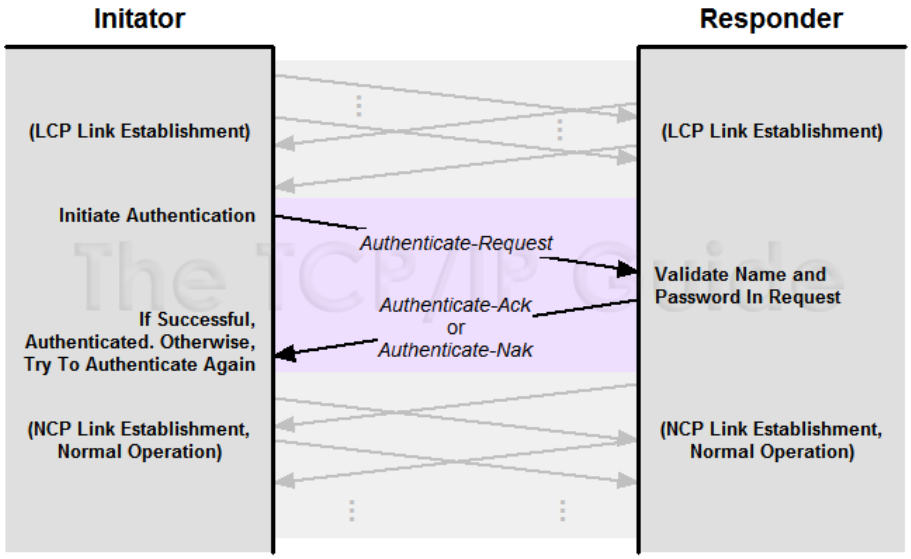
KLÍČOVÁ SLOVA

PAP, SSL

4.1 Password authentication protocol (PAP)

Využíván v protokolu PPP pro jeho autentizaci. PAP data nešifrovaná(dají se odchytit).
Ověřování v PAP protokolu je velice přímočaré, je složen ze dvou kroků:

- Žádost o ověření: odesílatel pošle zprávu Authenticate-Request, v kterém je obsaženo jméno a heslo, příjemci.
- Odpověď na ověření: příjemce žádosti zkontroluje jméno a heslo a rozhodne zda dále pokračovat v komunikaci. Pokud ano odešle zpět zprávu Authenticate-ACK. Pokud je tomu tak v opačném případě odešle Authenticate-Nak.



Obr. 4.1: PPP Password Authentication Protocol (PAP) Authentication

4.2 Secure Sockets Layer (SSL)

Protokol mezi transportní a aplikační vrstvou, zajišťující šifrování a autentizaci obou stran. Protokol je především využíván v HTTPS. Náplň této podkapitoly je obsažena v A.13.

SHRNUTÍ

V rámci této kapitolu se čtenář dozví nejpoužívanější protokoly relační vrstvy. V dalších částech je popsána problematika SSL.

OTÁZKY

1. Pro jaký protokol je využíván Password authentication protocol?
2. Popište jak funguje Password authentication protocol.
3. Pro jaký protokol je využíván Secure Sockets Layer?
4. Popište jak funguje Secure Sockets Layer.

MÍSTO PRO VAŠE POZNÁMKY

5 Připojení ke vzdálené konzoli

CÍLE KAPITOLY

Seznámení se s možnostmi vzdáleného připojení k zařízení na jeho konzoli. Seznámení se s výhodami a nevýhodami jednotlivých protokolů.

KLÍČOVÁ SLOVA

TELNET, RSH, SSH

5.1 TELNET

Protokol Telnet spadá do rodiny protokolů TCP/IP, jedná se o jeden z nejvíce historický významných protokolů. V raných dobách počítačových sítí, byl jedním ze základních problémů umožnit práci na vzdáleném zařízení. Protokol vyvinut s tímto cílem byl nazván Telnet.

Přesto že v současné době, uživatelé nikdy nevyvolaly protokol Telnet přímo, používaní některé z jeho principů nepřímo. Např. při posílání e-mailu, načtení webové stránky či FTP přenosu se stále používají technologie založené na Telnetu.

Telnet funguje na bázi klient-server, který je celou dobu spojen pomocí TCP a naslouchá na 23 portu. TCP udržuje spojení po celou dobu trvání relace a zaručuje, že data jsou přijímána spolehlivě a ve správném pořadí. Neboť TCP je full-duplex protokol, klient i server mohou posílat data dle libosti.

Pro uživatele po použití relace Telnet je stejné jako sedět přímo u terminálu vzdáleného hostitele. Spojení relace serveru s uživatelem začíná tak, že server požaduje přihlášení (jméno, heslo). Za předpokladu, že informace o přihlášení jsou platná, je uživatel přihlášen a může používat hostitele tak jak je nastaveno jeho oprávnění.

Více o Telnet technologii se můžete dočíst na The TCP/IP Guide.

5.2 Remote Shell (RSH)

Berkeley login příkaz (rlogin) umožňuje uživateli vzdálený přístup k UNIXovému hostiteli pomocí vnitřní sítě TCP/IP. Přihlášení pomocítoho to protokolu však mělo jednu nevýhodu a to pokud uživatel potřeboval zadat pouze jediný příkaz. V takovém případě byl nucen se přihlásit zadat příkaz a následně se zase odhlásit. Pro vetší pohodlí byl v důsledku vytvořen, jako variace rlogin, remote shell (rsh), který umožňuje vzdálený přístup k hostiteli a spuštění jednoho příkazu bez nutnosti se přihlásit a odhlásit.

RSH je nejužitečnější společně s automatickým přihlášením, takže program lze spustit bez nutnosti zadávat uživatelské jméno a heslo.

5.3 Secure Shell (SSH)

SSH byl navržen jako náhrada za telnet. Jeho výhodou oproti telnetu je šifrovan přenos. Náplň této podkapitoly je obsažena v A.12.

SHRNUTÍ

Náplní této kapitoly je seznámení s technologiemi vzdáleného připojení k počítači. Budou nastíněny problémy se zvýšenou mírou rizika, na kterou je nutné odpovědět dostatečným zabezpečením.

OTÁZKY

1. Mezi jakými protokoly/aplikacemi si můžeme zvolit pro připojení ke vzdálené konzoly?
2. Jaký je rozdíl mezi jednotlivými protokoly a jaké mají výhody či nevýhody?
3. Popište Telnet.
4. Popište SSH.

MÍSTO PRO VAŠE POZNÁMKY

6 Připojení ke vzdálenému GUI

CÍLE KAPITOLY

Seznámení se s možnostmi vzdálené správy počítače, pomocí protokolu VNC a RDP.

KLÍČOVÁ SLOVA

VNC, RDP

6.1 Virtual Network Computing (VNC)

Bavíme-li se o VNC, jedná se o program umožňující vzdálené připojení a správu počítače s grafickým uživatelským rozhraním. Za vývojem VNC stojí firma ORL (Olivetti & Oracle Research Lab). V roce 2002 byl výzkum ukončen a po uzavření ORL na vývoji pokračoval nově vzniklý projekt RealVNC. Nezávisle s tím to projektem vznikli ještě jiné implementace VNC, a proto je možné se setkat s různými typy VNC.

Jelikož celý systém je na bázi klient-server tak se VNC skládá ze serveru, klienta a komunikačního protokolu. VNC je není závislí na použité platformě, takže VNC klient se systému z Linuxové distribuce se může připojit třeba k VNC serveru se systémem Windows. Mezi další výhody patří například vícenásobné připojení klientu k jednomu serveru, to ovšem se v praxi dá použít spíše jako ukázka ostatním klientu, kdy jeden klient ovládá prostředky vzdáleného počítače a ostatní klienti mohou jen koukat. Jednou z dalším možností je nastavit klienta a server tak aby bylo umožněno připojení pomocí webového prohlížeče (nutná podpora Javy). Komunikace mezi VNC serverem a VNC klientem probíhá pomocí Remote Framebuffer (RFB) protokolu.

6.1.1 Zabezpečení

Je-li nastaveno zabezpečení musí klient projít autentizací ještě před zahájením komunikace. Autentizace je řešena na náhodné výzvě a ověřované odpovědi. Pokud jsou informace shodné dojde k navázání spojení klienta se serverem. Tento typ se nazývá challenge-response. Jeli klient spojen a již nastala samotná komunikace, stéle se jedná o nezabezpečenou komunikaci a je možné třetí stranou odchytit tuto sítovou komunikaci. Pokud chceme aby naše komunikace byla šifrovaná je nutné komunikovat pomocí VPN či zabezpečeného SSH tunelu.

6.2 Remote Desktop Protocol (RDP)

Stejně tak jako VNC, RDP slouží pro vzdálené připojení a správu počítače s grafickým uživatelským rozhraním. Tento protokol byl vyvinut firmou Microsoft a byl, již poprvé implementován ve Windows NT 4.0 Terminal Server Edition. Protokol je založen na základě sady T.120 agentury ITU (International Telecommunication Union). Tato sada je složena z komunikačních protokolů aplikační vrstvy umožňující real-time služby, vícebodové přenosy multimediálních dat. RDP je založen na architektuře klient-server.

6.2.1 Navazování spojení

Na samém začátku navázání spojení je nutno upřesnit společné nastavení mezi klientem a serverem, k tomu to slouží připojovací sekvence. Ta je rozdělena do deseti fází.

1. Inializace spojení
Klient odesílá Connection Request PDU na server. Server potvrdí spojení pomocí Connection Cornfirm PDU.

2. Nastavení komunikace
Pomocí MCS Connect Initial PDU a MCS Connect Response PDU si server s klintem vymění základní informace.

SHRNUTÍ

Náplň této kapitoly je téměř shodná s tou předchozí. Jedniným rozdílem je, že v předchozí kapitole byl nastíněn vzdálený přístup v textovém režimu. V této kapitole se zabýváme grafickým rozhraním.

OTÁZKY

1. Co je RDP?
2. Popište rozdíly mezi VNC a RDP.

MÍSTO PRO VAŠE POZNÁMKY

7 Sdílení dat

CÍLE KAPITOLY

Seznámení s možnostmi sdílení dat v síti a pochopení jak jednotlivé protokoly fungují.

KLÍČOVÁ SLOVA

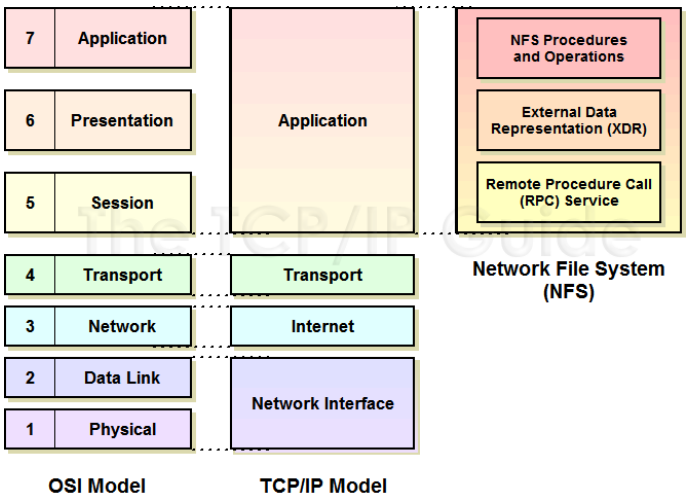
FTP, NFS, SAMBA

7.1 File Transfer Protocol (FTP)

File Transfer Protocol využívá protokolu TCP který spadá pod TCP/IP a může být použit bez ohledu na použitém operačním systému. Definován již v roce 1985. FTP využívá 20 a 21 portu. Náplň této podkapitoly je obsažena v A.11.

7.2 Network File System (NFS)

Protokol Sloužící pro vzdálený přístup k souborům. Byl vyvinut v 1984 Společností Sun Microsystems, ale v současnosti za jeho vývojem je Internet Engineering Task Force (IETF). Využívá především UDP protokolu. S verzí 3 bylo umožněno využívat i TCP protokol. NFS byl původně vyvinut pro operační systém UNIX. Provoz NFS je reprezentován pomocí tří základních složek :



Obr. 7.1: NFS Architectural Components

- Remote Procedure Call (RPC): pracuje na relační vrstvě a tato služba se používá k implementaci klient-serve.
- Externí datová reprezentace (XDR): je popisný jazyk, který umožňuje dotaovým typům bt definovány konzistentním způsobem. XDR je na úrovni prezentační vrstvy a umožňuje výměnu dat pomocí NFS mezi počítači přesto, že mají odlišné způsoby ukládání dat.
- NFS postupy a operace: jedná se o souhrn postupů a operací na aplikační vrstvě. Tyto postupy a operace definují konkrétní úkony, které mají být vykonány se soubory.

Tyto tři klíčové ”podprotokoly” tvoří převážnou část NFS. Podrobněji rozebrané je můžeme nalézt zde The TCP/IP Guide.

7.3 Samba (SMB)

Jedná se o implementaci síťového protokolu Server Message Block (SMB), sloužící převážně pro přístup ke sdíleným souborům v systému Microsoft Windows.

SHRNUTÍ

Posluchači bude v této kapitole obeznámen se základními možnostmi sdílení dat v LAN sítích. Ruku v ruce s tím jsou probrána bezpečnostní úskalí, rizika, spojená se sdílením dat.

OTÁZKY

1. Popište protokol FTP.
2. Co je Samba?

MÍSTO PRO VAŠE POZNÁMKY

8 Webové služby

CÍLE KAPITOLY

Cílem této kapitoly je pochopení problematiky HTTP a HTTPS protokolů, které jsou umístěny v aplikační vrstvě referenčního modelu ISO/OSI

KLÍČOVÁ SLOVA

HTTP, HTTPS

Náplň této kapitoly je obsažena v příloze A.9.

SHRNUTÍ Posluchač bude schopen chápat problematiku komunikace HTTP a HTTPS protokolů.

OTÁZKY

1. Popište hlavičku HTTPS protokolu.

MÍSTO PRO VAŠE POZNÁMKY

9 Emailová komunikace

CÍLE KAPITOLY

Seznámení se s jednotlivými protokoly sloužící pro e-mailovou komunikaci a jejich funkčností.

KLÍČOVÁ SLOVA

POP3, SMTP, IMAP

9.1 Post Office Protocol (POP)

Náplň této podkapitoly je obsažena v příloze A.4.

9.2 Simple Mail Transfer Protocol (SMTP)

Náplň této podkapitoly je obsažena v příloze A.5.

9.3 Internet Message Access Protocol (IMAP)

Náplň této podkapitoly je obsažena v příloze A.6.

SHRNUTÍ

Posluchač bude seznámen s komunikačními protokoly pro emailovou komunikaci, které se nacházejí na 4. vrstvě referenčního modelu ISO/OSI.

OTÁZKY

1. Popište protokol IMAP.
2. Popište protokol POP.
3. Popište protokol SMTP.

MÍSTO PRO VAŠE POZNÁMKY

10 Protokoly pro komunikaci v reálném čase

CÍLE KAPITOLY Cílem této kapitoly je, aby posluchač porozuměl základním principům tzv. IM (z angl.: Instant Messeaging)

KLÍČOVÁ SLOVA

IRC, Jabber

10.1 Internet Relay Chat Protocol (IRC)

V roce 1988 poprvé sepsal Finem Jarkko Oikarinen klient-server software sloužící pro komunikaci (chat). Protokol byl definován v RFC 1459, Internet Relay Chat Protocol a publikován až v roce 1993. V roce 2000 došlo k přepracování a rozšíření.

IRC protokol není založen na standardním klient-server modelu TCP/IP protokolu. IRC servery jsou TCP/IP zařízení na kterých běží IRC serverový software, ty jsou nastaveny s informacemi,které jim umožňují TCP spojeni na všechny ostatní. TCP je použit z důvodu spolehlivosti doručení dat. Serverové připojení slouží pro výměnu informací a uživatelských dat, které vytváří logickou IRC síť na úrovni aplikace. Páteřní IRC síť tvoří IRC komunikační služby. Uživatel spuštěním IRC klienta může přistupovat do sítě. Uživatel zadá název serveru v síti a naváže spojení pomocí TCP, to zajistí připojení do celé sítě a je schopen přijímat a odesílat zprávy ostatním uživatelům připojeným na stejný server ale i uživatelům připojeným k ostatním serverům.

Podrobněji je toto téma rozebrané na The TCP/IP Guide.

10.2 Jabber

V roce 1998 vznikl open-source projekt komunikačního serveru pojmenován Jabber. Dále pak v roce 2004 byl na základě použitého protokolu vytvořen standard XMPP (Extensible Messaging and Presence Protocol), což je ”rozšiřitelný protokol pro posílání zpráv a zobrazení stavu”. Původní protokol vznikl pro komunikaci v reálném čase, ale ukázalo se, že může být využit i ke vzdálené komunikaci programů nebo ovládání různých automatických služeb.

Síť Jabber/XMPP je decentralizovaná a skládá ze serveru, ke kterému se připojují klienti. Klienti můžou posílat zprávy ostatním klientům. Zpráva je odevzdána spolu s cílovou adresou na server, ke kterému je přihlášen, a ten jí pak doručí. Pokud je druhý z klientu na stejném

serveru je zpráva doručena ihned, je-li druhý z klientů na jiném serveru musí se zpráva nejprve dostat na správný server. Pokud není cílový klient přihlášen (online), zpráva čeká na serveru. Kdokoliv si může zřídit vlastní server a stále bude moc komunikovat s klienty připojených na jiných serverech. Uživatel si tak může zvolit server jaký chce, a který mu nabídne lepší služby. Jako identifikátor uživatele slouží Jabber ID (někdy také JID). Jedná se o řetězec složený z uživatelského jména, oddělovacím znakem @ a a názvem serveru. Např. franta@server.cz.

Příklad komunikace byl přebrán z wikipedia.org.

Co se děje při komunikaci mezi dvěma uživateli, ukazuje názorný příklad. Uživatelka Julie má účet na serveru Kapuletova.cz, takže její JID má tvar julie@kapuletova.cz. Chce si povídat s Romeem, jehož JID je romeo@montek.com. Když Julie napíše zprávu a pošle ji Romeovi, provede se několik akcí:

- XMPP klient Julie pošle její zprávu serveru Kapuletova.cz.

  - Pokud je Montek.com blokován, zpráva je smazána (zpět je zasláno chybové hlášení).

- Server Kapuletova.cz otevře spojení k serveru Montek.com a předá mu zprávu.

- Server Montek.com doručí zprávu klientovi Romea.

  - Pokud je server Kapuletova.cz na Montek.com blokován, zpráva bude smazána (zpět je zaslána chybová hláška).

  - Pokud není Romeo právě připojen, zpráva se uschová (na montek.com) a bude doručena při nejbližší příležitosti.

Protokol XMMP je založen an XML a komunikace je zajištěna pomocí TCP spojení. Detailní příklad přihlášení a komunikace můžeme nalezneme v článcích Seriál Jabber - komunikační protokol na root.cz.

SHRNUTÍ

Posluchač získá celkový přehled v oblasti IM.

OTÁZKY

1. Popište protokol IRC.

MÍSTO PRO VAŠE POZNÁMKY

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

11 Konfigurační protokoly sítě

CÍLE KAPITOLY

Posluchač bude seznámen s konfiguračními protokoly internetové sítě. Dále pochopí funkci a nasazení konfiguračních protokolů.

KLÍČOVÁ SLOVA

BOOTP, DHCP, DNS

11.1 Bootstrap Protocol (BOOTP)

Na začátku 90 let by sítě poměrně malé a jednoduché, tudíž nepotřebovali automatizovat konfiguraci parametru pro IP hosty. To se ale s modernizací a stále se rozšiřujícími sítěmi brzy měnilo a bylo potřeba zajistit automatizaci. Nadále byla automatizace potřeba ke konfiguraci zařízení jako třeba bezdiskových stanic. První pokusem o vyřešení takzvaného "bootstrap" problému byl Reverse Address Resolution Protocol (RARP). Byl vytvořen v roce 1984, jednalo se o přímou adaptaci low-level Address Resolution Protocol (ARP), který váže IP adresy k hardwarovým adresám. Potíže s RARP, byly jeho omezení. BOOTP byl vytvořen v roce 1985 a řešil ostatní problémy RARP.

BOOTP je klient-server protokol, implementován jako softwarový protokol vyšší vrstvy, využívající UDP k přenosu zpráv. Začal podporovat odesílání více informací o konfiguraci než pouze IP adresu.

11.1.1 BOOTP přehled, historie a standardy

BOOTP se používá v první fázi Bootstrappingu

Musíme poznamenat, že přestože jméno BOOTP naznačuje, že obsahuje vše potřebné k bootování zařízení bez paměti, není to pravda. Jak již BOOTP standard popisuje, je třeba dvou fází. V té první, je klientu poskytnuta adresa a další parametry. Ve druhé fázi, klient stáhne software, operační systém a drivery, které mu umožní pracovat v síti a provádět úkony. BOOTP se stará pouze o první z těchto dvou fází: přidělení adresy a konfigurace. Druhá fáze je vykonávána jednoduchým transportním protokolem jako je TFTP.

Když byl BOOTP vytvořen, rozhodli se tvůrci zahrnout do jeho implementace možnost určení parametrů výrobcem. Jak se TCP/IP stávalo více a více komplexnější, přišlo se na to, že tato oblast může být využita pro určitou metodu komunikace, za pomocí parametrů, které jsou běžně vyžadovány hostem a jsou nezávislé na výrobci. Toto bylo poprvé navrženo v RFC 1048, BOOTP Vendor information extensions (rozšíření informací uvedených výrobcem) v únoru 1988.

Fakt, že BOOTP může být použit k poskytování jiných informací klientovi než jen IP adresy, může být výhodné, když klient už svou IP adresu zná. BOOTP může být také použit tak, že předá parametry, které chce, aby měli všechny stanice, aby si byl jistý, že využívají síť konzistentně. Také v případě, že zařízení mají nějakou paměť (tudíž není třeba používat BOOTP protokol na získání IP adresy) BOOTP může být využit pro získání jména boot souboru pro 2. fázi bootstrappingu.

Změny BOOTP a vývoj DHCP

BOOTP byl TCP/IP protokol sloužící pro konfiguraci od poloviny 80. do konce 90. let. RFC 1048 byly nahrazeny dalšími normami (1084,1395 a 1497). Některé nejasnosti se objevili v předpisu RFC 951 a to jak některé části protokolu mají být interpretovány a jak určité postupy v BOOTP fungují.

RFC1542, Klasifikace a Rozšíření pro Bootstrap Protokol byly publikovány v říjnu 1993, které se tímto tématem zabývaly a upravovaly protokol. Ve skutečnosti je RFC 1542 pouze opravou RFC 1532, které vykazovalo drobné chyby. Ačkoli byl BOOTP docela úspěšný, měl sám o sobě určité slabiny. Jednou takovou slabinou byla absence podpory dynamického přidělování adres. Toto bylo obzvláště živé téma na konci 90. let, kdy Internet zažil boom. To vedlo k vývoji DHCP. To že DHCP převzalo prim jako TCP/IP protokol neznamená, že BOOTP úplně vymizel. Stále se v některých sítích používá. Navíc, DHCP je přímo postaveno na principu BOOTP, mají společných mnoho vlastností, jako je společný formát zpráv. BOOTP vendor rozšíření byly použity jako základ DHCP nastavení, které pracují na stejném principu, ale mají vlastnosti navíc.

BOOTP Klient/server posílání zpráv a adresování

BOOTP může být použit na velké škále zařízení, ale hlavní myšlenou při vývoji bylo, aby protokol dokázal automaticky nakonfigurovat ”hloupá” zařízení, která nemají vlastní paměť. Většina z těchto zařízení jsou poměrně hodně limitována svými možnostmi. BOOTP je tak ideálním nekomplikovaným řešením, které zařídí konfiguraci bez zbytečně komplikované implementace.

BOOTP Klienti a Servery

Jako jiné TCP/IP protokoly, BOOTP je typu klient/server. Základní operace toho protokolu je výměna zpráv mezi klientským BOOTP a BOOTP na serveru. BOOTP klient může být jakékoli zařízení nastavované pomocí BOOTP protokolu. BOOTP server je síťové zařízení, které bylo speciálně nastavené (navržené) k tomu, aby odpovídalo na požadavky klientů.

Transport a výměna zpráv

Jako komunikační kanál využívá BOOTP User Datagram Protocol (UDP) jako protokol 4 vrstvy a to z několika důvodů. Zaprvé UDP není tak náročný jako další přenosový protokol na 4 vrstvě (TCP) a je ideální pro komunikaci typu dotaz/odpověď. Za druhé, BOOTP klient nezná adresu serveru, tak je požadavek vysílán na broadcastu po celé lokální síti, UDP podporuje broadcast zatímco TCP ne.

UDP používá speciální rezervovaný port pro BOOTP servery: UDP port 67. BOOTP servery ”poslouchají” tento port, aby zachytili požadavky od klientů. Po vyřízení dotazu, server pošle odpověď zpět na klienta. Způsob zasílání odpovědí na klienta záleží na tom, zda klient zná svoji adresu.

- Klient zná svoji adresu: Jsou některé případy, kdy klient zná svou adresu, server ji tedy využije a zašle odpověď přímo.

- Klient nezná svoji adresu: BOOTP se samozřejmě často používá k tomu, aby přiřadil IP adresu zařízení, které svou adresu nezná. To se často označuje jako problém "slepice a vejce", který reprezentuje nekonečnou smyčku. Aby BOOTP vyřešil toto dilema, má dvě možnosti. Jestli to podporuje operační systém, server může použít klientovu fyzickou adresu pro vytvoření ARP vstupu pro zařízení (Address Resolution Protocol) a poté použije unicast 2. vrstvy pro odeslání odpovědi. Jinak bude vysílat odpověď broadcastem.

Použití Broadcastu a portů

Fakt, že BOOTP servery mohou využívat broadcast k posílání zpráv zpět na klienta, vyžaduje malou změnu od klasického přístupu jak TCP/IP využívá klientské porty. Vzpomeňme si, že normálně klient ve vztahu klient/server pro výměnu pomocí UDP nebo TCP generuje dočasný port, který pak využívá jako zdrojový port ve svém požadavku vyslaném na server. Server pak posílá odpověď zpět na adresu klienta za pomocí tohoto dočasného portu. Toto dočasné číslo musí být unikátní pro každou IP adresu, ale nemusí být unikátní pro všechny zařízení v síti. Například zařízení A může používat dočasné číslo portu 1248 pro http požadavek na Web server, zatímco zařízení B může používat port 1248 pro zaslání DNS požadavku.

Jakmile BOOTP server používá broadcast, nezaměřuje se cíleně na jedno zařízení pomocí unicastu. To znamená, že nemůže bezpečně komunikovat za pomoci dočasného portu: některé zařízení v síti, si mohlo přivlastnit stejné dočasné číslo portu, a může si splést odpověď, která nebyla určena jemu. Aby se vyhlo tomuto problému, další velmi známý port je použit exkluzivně pro klienty – UDP port 68. Klient poslouchá tento port pro zachycení odpovědi na broadcastu nebo unicastu, zatímco zařízení, která dotaz na server neposlala, ho budou ignorovat. Tento proces "dvojitého broadcastu" je zobrazen na obrázku 254.

BOOTP je relativně jednoduchý protokol typu klient/server, který využívá broadcast ke komunikaci se zařízeními které nemají přidělenou IP adresu. V tomto příkladu, se zařízení A snaží zjistit svou IP adresu a další parametry. Vysílá broadcastem požadavek na lokální síť s využitím UDP portu 67 a poslouchá odpověď na portu 68. Zařízení D je nakonfigurovaný BOOTP server a poslouchá požadavky na portu 67, jakmile požadavek zachytí a zpracuje, odpověď pošle na port 68 s odpovědí, jaká je IP adresa zařízení, které dotaz vyslalo.

Přesměrování ztracených zpráv

Nevýhoda použití tak jednoduché metody UDP pro BOOTP komunikaci je, že nemůžeme zajistit kvalitu komunikce. UDP je nespolehlivá metoda přenostu, to znamená že BOOTP požadavek může být vtažen do černé díry UDP přenosu a už nikdy se neobjevit, to samé platí pro odpověď ze serveru, může se ztratit dřív, než dojde ke klientovi. Tak jako jiné protokoly využívající UDP, BOOTP klienti to řeší za použití timeru, který pošle zprávu znovu po uplynutí určitého času.

Ale BOOTP klient se o to jak implementuje strategii zasílání opakovaných žádostí, musí postarat sám. Vezměme si příklad, kde síť s 200 BOOTP klienty ztratí elektřinu. Jakmile je proud obnoven, všechny zařízení se restartují a pokusí se poslat BOOTP požadavek ve zhruba stejnou chvíli. Je velmi pravděpodobné, že kvůli velkému množství požadavků budou některé ztraceny, nebo bude server zahlcen a některé požadavky nevyřídí a zahodí je.

Jestliže všichni klienti budou nastaveni stejně, po uplynutí určité doby se pokusí vyslat znovu žádost na server a problém se objeví znovu. Aby se tomu vyhnulo, BOOTP standard doporučuje použít dynamickou změnu intervalu posílání požadavků, začínající na intervalu 4s a pro úspěšnou komunikaci ji pokaždé zdvojnásobit. Náhodná doba vysílání požadavku je také implementována do tohoto řešení, aby se omezili kolize požadavků. Velmi podobný princip funguje na Ethernetu.

Například, první znovuzaslání žádosti se objeví mezi 0-4 sekundou, druhé znovuzaslání, pokud je potřeba, bude vysláno v intervalu 0-8s +- a tak dále. Tato metoda pomáhá redukovat šanci, že požadavek bude znovu zahozen, nebo se ztratí, také to pomáhá zajistit, že přenos na BOOTP nezahltí síť.

11.1.2 BOOTP detailní pohled na operace

Teď, když víme, jak funguje výměna zpráv na BOOTP protokolu, pojďme se blíže podívat na operace protokolu. Toto nám přiblíží postup, jak klient a server zpracovává zprávy a také nám to osvětlí některé důležité aspekty BOOTP formátu zprávy. Pochopení základních operací BOOTP bude velmi užitečné, až budeme zkoumat přenos žádostí pomocí relay agentů a až budeme probírat DHCP.

BOOTP BOOTSTRAPPING postup

Klient
1. generuje BOOTREQUEST zprávu
2. vysílá BOOTREQUEST zprávu
Server
3. Přijme a zpracuje BOOTREQUEST zprávu
4. Vygeneruje BOOTREPLY zprávu
5. Pošle BOOTREPLY zprávu
Klient
6. Přijme a zpracuje BOOTREPLY zprávu
7. Dokončí bootstrapping proces například pomocí protokolu TFTP.

1. Klient vytvoří požadavek

Klientské zařízení započne komunikaci vytvořením BOOTP žádosti. Během vyplňování požadavku uvede tyto údaje:

- Nastaví typ zprávy (Op) na hodnotu 1, pro BOOTRREQUEST zprávu
- Pokud zná svojí IP adresu, kterou bude používat, uvede ji v poli CIAddr. Pokud adresu nezná, vyplní pole nulami.
- Přiloží vlastní adresu 2. vrstvy do pole CHAddr. Toto je použito serverem, aby dokázal určit správnou adresu a další parametry klienta.
- Generuje náhodný identifikátor pro zprávu a vloží ho do XID pole.
- Klient může specifikovat, na který konkrétní server chce požadavek vyslat (položka SName) a může si dokonce určit o jaký specifický boot file chce server požádat. Toto uvede v poli File.

2. Klient posílá požadavek

Klient vyšle požadavek BOOTREQUEST tak, že ho vyšle na adresu 255.255.255.255. Nebo pokud zná konkrétní adresu BOOTP serveru, může použít unicast.

3. Server přijme požadavek a zpracuje ho

BOOTP server, který poslouchá na portu 67, přijme požadavek vyslaný broadcastem a zpracuje ho. Pokud bylo uvedeno jméno jiného serveru, než je ten, který ho přijal, server

může požadavek zahodit. To se stává především, jestliže server ví, že server, který si klient vyžádal, je také na lokální síti. Pokud se opravdu jedná o vyžádaný server, nebo jeho jméno nebylo specifikováno, server žádost vyřídí.

4. Server vytvoří odpověď

Server vytvoří odpověď tak, že zkopíruje celý požadavek a změní několik údajů:

- Změní typ zprávy (Op) na hodnotu 2, pro BOOTREPY zprávu
- Vezme specifikovanou adresu z CHAddr a použije ji pro vyhledání IP adresy ve své tabulce. Nalezenou hodnotu poté vloží do pole YIAddr (your IP address).
- Prohlédne pole Field a připojí požadovaný typ. Pole nechá prázdné, pokud nebylo vyplněno v žádosti.
- Připojí vlastní IP adredu a jméno do SIAddr a SName.
- Nastaví nějaké hodnoty do Vend pole. (NEMAM TUŠENÍ CO TO JE)

5. Server pošle podpověď

Server zašle odpověď v závislosti na obsahu zprávy:

- Jestliže byl nastaven příznak B (broadcast), nemůže být klientovi zpráva doručena pomocí unicastu, takže server použije broadcast.
- Jestliže je pole CIAddr nepprázdné, server pošle odpověď unicastem na CIAddr adresu
- Jestliže je příznak B prázdný a pole CIAddr je také prázdné, server použije ARP nebo broadcast.

6. Klient přijme a zpracuje odpověď

Klient přijme odpověď serveru a uloží si informace obsažené ve zprávě.

7. Klient poté dokončí BOOT proces

Jakmile je klient nakonfigurován, přejde do druhé fáze bootrstapping procesu, použitím protokolu např. TFTP kdy si stáhne boot soubor obsahující software operačního systému, za použití jména Filename, které mu poskytl server.

Interpretace Klientské IP adresy (CIAddr)

Komplikace můžou nastat, pokud klient specifikuje IP adresu v CIAddr poli. Jak přesně interpretovat toto pole.

- Znamená to, že klient už používá tuto IP adresu?
- Nebo se jedná o adresu, kterou měl přiřazenou, když naposledy bootoval?
- Také zde vyvstává problém: Co dělat, když server přiřadí adresu zařízení do YIAddr která bude jiná, než používá klient?
- Měl by server přepisovat klientovu adresu?
- Nebo by to měl klient ignorovat?

- Kdo rozhodne, server nebo klient?

Velmi mnoho nejasností a nejednoznačnosti se promítlo do standardu, který BOOTP používá, což vedlo k různým přístupům implementace řešení. Objevili se některé implementace, které používali CIAddr ve smyslu ”klient požaduje tuto IP adresu”, což původně nikdy nebylo součástí BOOTP. Tato metoda není vůbec vhodná, jednoduše proto, že vede přímo k tomu, že BOOTP odpovědi nikdy nedojdou ke klientovi.

RFC 1542 byla vytvořena zláště proto, aby ujasnila pravidla zacházení s CIAddr a YIAddr poli:

- Jestliže je klient ochotný přijmout jakoukoli adresu, kterou mu server nabídne, klient nastaví pole CIAddr na nuly, i přesto, že zná předchozí adresu.
- Jestliže klient vyplní adresu, znamená to, že tuto adresu bude používat a musí být připraven přijmout zprávy vyslané unicastem na tuto adresu.
- Jestliže klient specifikuje adresu v CIAddr a dostane jinou adresu od serveru v poli YIAddr, adresa poskytnutá serverem je ignorována.

Nezapomeňme, že ne všechen hardware musí nutně používat tuto interpretaci poskytnutou RFC 1542, takže se mohou stále vyskytnout problémy se starším hardwarem. RFC 1542 byla ale napsána v roce 1993, takže se s tímto problémem již téměř nesetkáme.

11.2 Dynamic Host Configuration Protocol (DHCP)

Náplň této podkapitoly je obsažena v příloze A.8.

11.3 Domain Name System (DNS)

Tato služba slouží pro převod z jmenné adresy (doménová) na číselnou (IP). V praxi to vypadá takto pokud, chceme v prohlížeči přejít na nějakou stránku, napíšeme její název např. www.seznam.cz (doménová adresa) požadavek se odešle na DNS server, ten zašle zpět klientovy IP adresu 77.75.79.39, a pomocí této adresy klient kontaktuje cílový počítač. Záznamy na DNS servech nemají v záznamech pouze IP adresy a názvy, ale také seznam služeb, které jsou na dané doméně k dispozici uživatelům. Domény můžeme dále rozdělit do tří skupin:

Doména nejvyšší úrovně

Pokud se budeme bavit o doméně nejvyššího řádu, je to tak zvaná Top Level Domain (TLD), neboli doména 1. řádů. Jedná se o příponu, která je připojena na konec doménového jména. Domény prvního řádu pak ještě můžeme dělit na národní (.cz, .ru, .sk, ...)a na nadnárodní (.com (commerce), .net (network), .org (organization), .edu (education), ...).

Doména 2. úrovně

Doména druhého řádu určuje přímo jedince, firmu či společnost. Napr. ujep.cz

Doména 3. úrovně (subdoména)
Subdoména pouze rozšiřuje doménový název o další úroveň. Např. ki.ujep.cz

11.4 Uniform Resource Locator (URL)

URL je přímá adresa (řetězec znaků), která slouží ke spicifikaci umístění zdrojů infprmací at už se jedná o nějaký dokument a nebo službu na internetu. URL obsahuje adresu serveru, protokol, pomocí kterého tam budeme přistupovat a umístění zdroje.
Např. http://physics.ujep.cz/ mskoumal/school/uvod-do-pocitacovych-siti-i-pdf.pdf
Dále je náplň této podkapitoly je obsažena v příloze A.7.

SHRNUTÍ Po prostudování této kapitoly bude posluchač schopen pochopit základní principy fungování konfiguračních protokolů sítě.

OTÁZKY

1. Popište co je URL.
2. Popište co je DNS.
3. Popište co je DHCP.

MÍSTO PRO VAŠE POZNÁMKY

12 Protokoly pro přenos a publikování multimédií

CÍLE KAPITOLY

Kapitola se zabývá protokoly, které jsou speciálně určeny pro přenos multimediálních dat.

KLÍČOVÁ SLOVA

UPnP (DLNA), RTP, RTSP, RTCP

12.1 Real-time Transport Protocol (RTP) - RTP Control Protocol (RTCP)

Náplň této podkapitoly je obsažena v materiálech [4, strna 149] kapitola 8.3.4 Protokol RTP.

12.2 Real Time Streaming Protocol (RTSP)

Náplň této podkapitoly je obsažena v materiálech [4, strna 142] kapitola 8.3.4 Protokol RTSP.

12.3 Universal Plug and Play (UPnP)

Architektura UPnP umožňuje bezdrátovým zařízením P2P spojení počítačů a síťových zařízení. Komunikace probíhá pomocí XML přes HTTP a TCP/IP spojení. Hlavních výhodami této technologie je, že nepotřebuje žádné dodatečné ovladače, je postavena na společných protokolech a je zcela nezávislá na operačním systému, programovacím jazyku a použitém médiu.

UPdP se skládá ze dvou typů systémů. Prvním z nich jsou Controllers (kontrolní body). O techto zařízeních můžeme uvažovat jako o master, ty mohou být zakončeny systémem nebo systémem s uživatelským rozhraním (např. PC). Druhým z nich jsou pak zařízení (Devices). Ty jsou řízeny pomocí controllers. Skládají se alespoň z jedné službya případných vnořených zařízení, které také mohou obsahovat zařízení či služby (např. Mikrovlná trouba obsahuje služby k přípravě pokrm, ale také může mít vnořené zařízení hodiny, které nám udávají čas).

Je složena z 3 UPnP vrstev a standardních protokolů:

- Vrchní vrstva: Vendor defined

  -Udávají dodavateli specifické informace (např. sériové číslo)

- Další dvě vrstvy: UPnP Forum Working Committee Defined a UPnP Device Architecture Defined

  -Udávají zařízení specifické globální informace (např. videorekordér info, DVD player info, atk.)

- Čtvrtá vrstva: SOAP, SSDP a GENA

  -Simple Object Access Protocol: používá se pro vzdálené volání procedur (RPC)

  Pracuje dobře s proxy a SSL

  -Simple Service Discovery Protocol: používá se pro vysílání a vyhledávání

  Pomáhá při přidání/odebrání zařízení ze sítě

  -Generic Event Notification Architecture: slouží k odběru a publikování zprávy

- Nejnižší tři vrstvy: TCP, UDP, IP, HTTP

  -UPnP postavené na protokolu TCP / IP

  -Zprávy dodávány pomocí HTTP

Protokol je rozdělen do šesti částí:

- **Adresování** - Získání adresy je možno buď pomocí DHCP serveru od kterého získá adresu a nebo pokud je DHCP server nedostupný použije Auto IP, ve kterém si systém vybere vlastní IP adresu z dané sady adres.

- **Zjišťování** - Controlleres hledají zařízení v zájmu a naopak. Pro vyhledání je použit SSDP broadcast. Dostanou pouze základní informace (Identifikátor zařízení a typ zařízení).

- **Popis** - Jelikož controller potřebuje více informací zařízení mu odešle informace pomocí XML. Funce zařízení, ID dodavatele, vestavěné služby, URL se společnými službami, dále pak všechny možné akce a argumenty spojené se službami a akcemi, stavové proměnné.

- **Ovládání** - Controller zná již všechny potřebné informace k ovládání zařízení. Pro ovládání zařízení odešle příkaz na specifickou URL. Zmeny stavů se projevují ve změně stavových proměnných. XML zprávy putují pomocí SOP protokolu.

- **Upozornění o události** - Zatím co je zařízení ovládáno je nutné odesílat zprávy o změnách. Události vyjádřené v XML jsou odesílány pomocí GENA.

- **Prezentace** - Pokud má zařízení prezentační URL, controller jej může zadat do vyhledávače a umožnímu to tak bud zařízení monitoroivat nebo přímo řídit.

SHRNUTÍ

Po prostudování bude posluchač schopen posoudit základní problémy komunikačních protokolů pro přenos multimediálních dat.

OTÁZKY

1. Popište co je RTP.
2. Popište co UPnP.

MÍSTO PRO VAŠE POZNÁMKY

SEZNAM OBRÁZKŮ

ODKAZY NA LITERATURU

[1] J. Jelínek: *Úvod do počítačových sítí I.* [online]. UJEP (2005).
Dostupné z: https://ki.ujep.cz/enastenka/Opory/Jelinek-uvod-do-pocitacovych-siti-i.pdf

[2] V. Valenta, P. Simr: *Počítačové sítě I.* [online], UCP UJEP v Ústí nad Labem (2015).
Dostupné z: https://ki.ujep.cz/enastenka/Opory/ValentaSimrKIPSI1PocitacoveSiteI.pdf

[3] Ing. Zdeněk Růžička: *Informační technologie* [online], Střední průmyslová škola strojní a elektrotechnická, Dukelská 13, České Budějovice (2014).
Dostupné z: https://ki.ujep.cz/enastenka/Opory/Ruzicka.pdf

[4] Prof. Dr. Ing. Zdeněk Kolka: *Počítačové a komunikační sítě* [online], FEKT Vysokého učení technického v Brně (2013).
Dostupné z: https://ki.ujep.cz/enastenka/Opory/PSI-Kolka.pdf

A Informační technologie - Zdeněk Růžička

Informační technologie

Střední průmyslová škola strojní a elektrotechnická,
Dukelská 13, České Budějovice

Ing. Zdeněk Růžička

- Vzdělávací oblast: Elektrotechnika
- Tématická oblast: Informační technologie
- Šablona: VY_32_INOVACE_IT.10.181
- Číslo materiálu: VY_32_INOVACE_EE.23.441
- Autor: Ing. Zdeněk Růžička
- Ročník, pro který je materiál určen: 4. ročník Elektrotechnika 26-41-M/01

Prohlašuji, že při tvorbě výukového materiálu jsem respektoval všeobecně užívané právní a morální zvyklosti, autorská a jiná práva třetích osob, zejména práva duševního vlastnictví (např. práva k obchodní firmě, autorská práva k software, k filmovým, hudebním a fotografickým dílům nebo práva k ochranným známkám) dle Zákona 121/2000 Sb. (Autorský zákon).

Nesu veškerou právní odpovědnost za obsah a původ svého díla.

Dále prohlašuji, že výše uvedený matriál jsem ověřil ve výuce a provedl o tom zápis do TK.

Dávám souhlas, aby moje dílo bylo k dispozici veřejnosti k účelům volného užití (§ 30 odst. 1. Zákona 121/2000 Sb.) tj., že k uvedeným účelům může být kýmkoliv zveřejňováno, používáno, upravováno a uchováváno.

Anotace

- Materiál je vytvořen jako prezentační doplněk výuky.
- Inovativní je způsob zpracování, které vede k rozšíření možností výkladu i mimo tabuli, s níž lze spolupracovat.
- Vzniká tím možnost většího zaujetí žáků a tím i k větší motivaci a osvojení učiva.

A.1 Aktivní síťové prvky

Aktivní síťové prvky obecně

- Pod pojem "*aktivní síťové prvky*" se v dnešní době zařazují všechna zařízení, která slouží potřebám vzájemného propojování v počítačových sítích.
  V prvním přiblížení si lze představit, že "*aktivní síťový prvek*" je všechno to, co nějakým způsobem **aktivně působí na přenášené signály**
  (zesiluje nebo nějak jinak modifikuje).

Aktivní síťové prvky – nejzákladnější zařízení

Mezi nejzákladnější aktivní síťová zařízení můžeme zařadit:

- **1. Repeater (opakovač)**
- **2. Hub (rozbočovač)**
- **3. Switch (přepínač)**
- **4. Bridge (můstek)**
- **5. Router (směřovač)**

Repeater

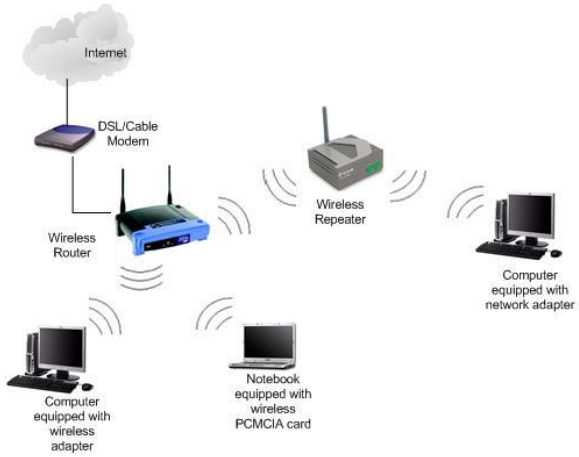
Repeater neboli opakovač je Elektronické zařízení, které funguje jako **zesilovač signálu**.

Toto zařízení nemá žádnou paměť - vše co přijme, zesílí a odešle "bez rozmyšlení" hned dál.

Repeater - použití

- Repeater používáme proto, že kvalita signálu se s narůstající vzdáleností ztrácí, proto signál "nastavujeme" repeatery, aby se "zopakovala" kvalita signálu, kterou měl na začátku a dosáhl tak delší vzdálenosti.

- Jeho další funkcí je čistění signálů od deformací a zkreslení .
- Pracuje na fyzické vrstvě modelu ISO

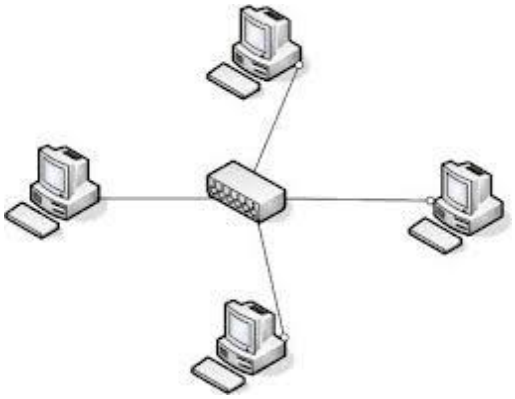


Hub

- Hub neboli rozbočovač je zařízení, které umožňuje **větvit počítačovou síť**, nejčastěji do hvězdicové topologie.
- Signál se kterým zařízení pracuje je rovněž zesílen.

Hub - použití

- Huby jsou dnes již na ústupu, jsou nahrazovány modernějšími a chytřejšími switchi, přesto mohou být huby v některých případech užitečné



- Např. pokud použijeme hub a propojíme dva porty, ochromíme pouze uživatele připojené k hubu, ne uživatele v celé síti (jak by se stalo v případě switche)
- Pracuje na fyzické vrstvě modelu ISO

Switch

- Switch neboli přepínač lze vnímat jako dokonalejší formu Hubu.
- Stejně jako v případě Hubu je elektronický signál zesílen, ale na rozdíl od Hubu jej ale **zasílá pouze na port, pro který je určen**.

Bridge

- Bridge neboli můstek je zařízení, které **fyzicky umožňuje spojení a následné sdílení informací** přes tzv. bránu (=gateway) mezi více počítačovými sítěmi.
- Dále také konvertuje data do formátu, se kterým je síť schopná pracovat.
- Pracuje na linkové vrstvě modelu ISO

Bridge – výhody a nevýhody

- Bridge jako takový má řadu výhod a nevýhod mezi které patří

- **Výhody:**

není ho potřeba konfigurovat,
snižuje velikost kolizní domény,
transparentní k protokolům z vyšších vrstev,
lacinější než router

- **Nevýhody:**

neomezuje rozsah všesměrového vysílání,
vyšší odezva, než opakovače (repeater) z důvodu čtení MAC adresy,
dražší než opakovače,
přemosťováním různých MAC protokolů dochází k chybám

Router

- Router neboli směřovač je aktivní síťové zařízení, které procesem zvaným routování přeposílá datagramy směrem k jejich cíli.
- Netechnicky řečeno, router spojuje dvě sítě a přenáší mezi nimi data

Router – popis činnosti

- Zařízení propojující sítě, které pracují se stejným síťovým protokolem. Přenáší pakety tou nejlepší možnou cestou k cílovému hostiteli. Oproti switchi je pomalejší - paket musí totiž nejprve načíst do své vyrovnávací paměti a až poté se rozhodne, co s ním bude dál dělat. Pracuje se 2 tabulkami - již zmíněnou **přepínací tabulkou** a nově se **směrovací tabulkou**, díky níž zná topologii sítí a může tak zvolit algorytmus přenosu dat. Pracuje na třetí vrstvě modelu ISO/OSI, neboť odesílá data podle IP adres - tedy protokolově v síti. Může pracovat buď na základě **statického směrování**, nebo na zakládě **dynamického směrování**. Statické směrování = směrovací tabulku vyplňuje uživatel, takže automaticky nereaguje na změny v síti. Naopak dynamické směrování - jak název napovídá - na změny v síti reaguje.

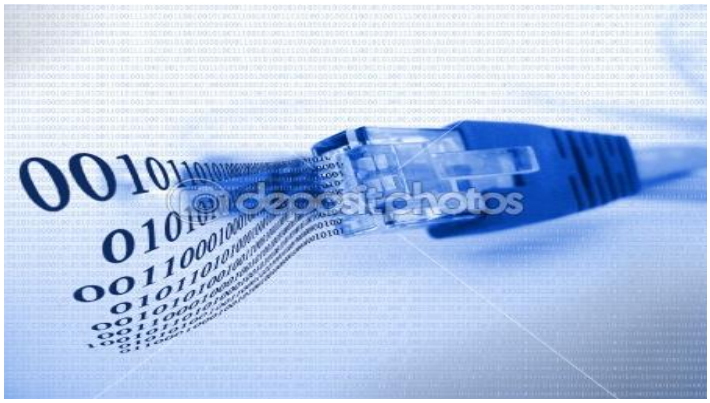
Zdroje informací:

- cs.wikipedia.org
- http://home.zcu.cz/~svidensb/
- obrázky vyhledávány pomocí google.cz

A.2 Pasivní síťové prvky

Pasivní síťové prvky strukturované kabeláže

- Základním kamenem počítačových sítí a celé ICT infrastruktury jsou pasivní prvky strukturované kabeláže, které zajišťují potřebné přenosy dat po tzv. fyzické vrstvě. Jedná se o kabeláž (vlastní kabelové vedení) a o technologii ukončení kabeláže (datové zásuvky, patch panely). Pasivní síťový prvek data pouze přenáší a nijak je neupravuje.
- Může se jednat o kabeláž metalickou nebo optickou. Základní vlastností strukturované kabeláže je její univerzálnost. Ta zaručuje použitelnost jednotné kabeláže jak pro potřebu přenosu dat a hlasu, ale také třeba i obrazu, zabezpečovacího systému a přenos informací pro měření a regulaci.

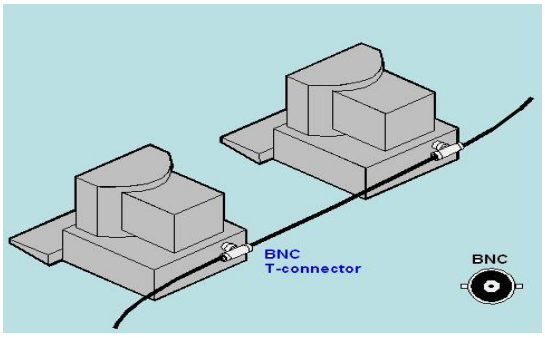


Metalické kabely

- **Koaxiální kabel**

Je tvořen dvěma vodiči - první tvoří středovou žílu a druhý tvoří izolační vrstvu kolem prvního. Celý kabel je ještě zvenku obalen plastovým obalem. Užívá se ve sběrnicové topologii. Lze jím přenášet stejnosměrný proud, stínit nízkofrekvenční signály, ovšem nejčastější funkcí koaxiálního kabelu je přenos elektromagnetického vlnění o vysokém kmitočtu. Existují dva druhy koaxiálního kabelu - **tlustý a tenký**.

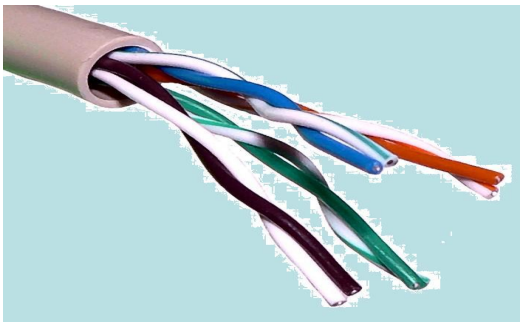
Tlustý: tloušťka kabelu je 0,5 palců, přenáší signál až do vzdálenosti 500 m.
Tenký: tloušťka kabelu je 0,25 palců, přenáší signál do vzdálenosti necelých 200m.



- Kroucená dvojlinka

Je tvořena 4 páry vodičů, ale 2 páry - modrý a hnědý - se požívají ve výjimečných případech. Důvodem kroucení vodičů je zlepšení elektrických vlastností kabelu. Minimalizují se takzvané přeslechy mezi páry a snižuje se interakce mezi dvojlinkou a jejím okolím, tj. je omezeno vyzařování elektromagnetického záření do okolí i jeho příjem z okolí. Oba vodiče jsou v rovnocenné pozici, a proto kroucená dvojlinka patří mezi tzv. symetrická vedení. Signál přenášený po kroucené dvojlince je vyjádřen rozdílem potenciálů obou vodičů.

Rozlišujeme tyto kabely na dva druhy –
UTP - Kroucená nestíněná dvojlinka
STP – Kroucená stíněná dvojlinka

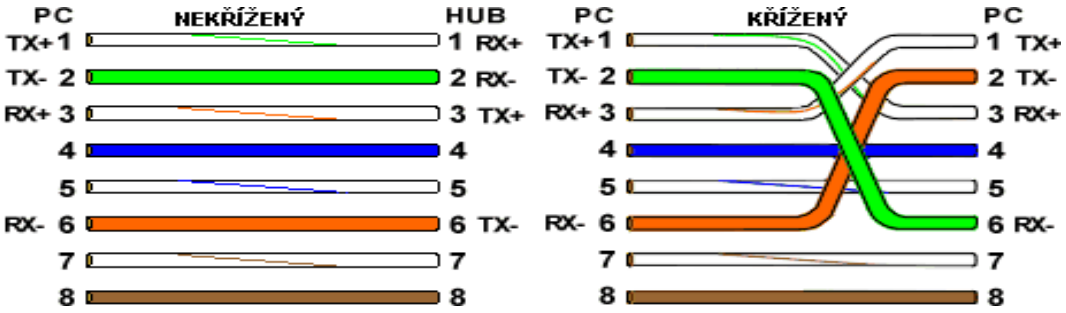


Koncovky Metalické Kabeláže

- **Koncovka RJ-45** je dnes nejčastěji používaný typ zapojení síťových kabelů UTP a STP. Mimo to se ale používá ke spojení xDSL modemů, ISDN zařízení, E1 atp. Vytlačila mnoho ostatních koncovek, z důvodu snižování počtu vodičů a modernizace počítačového vybavení. Je to koncovka typu 8P8C (z angličtiny: 8 pozic, 8 vodičů). RJ-45 může mít dvě podoby: samičí (zásuvka) nebo samčí.

Zapojení koncovek RJ45

- V zásadě se používají dva druhy zapojení kabelů UTP(popř. STP). A to:
- Křížené
- Nekřížené



- Křížený kabel se dříve užíval na přímé spojení mezi jednotlivými počítači, dnes se již užívat nemusí, jelikož dnešní síťové karty dokážou samy kabel rozpoznat a přislušně upravit vysílací a přijímací kanál.

Optické kabely

- Optická kabeláž je nedílnou součástí metalické strukturované kabeláže a používá se všude tam, kde metalická kabeláž již nevyhovuje podmínkám přenosu dat. Jedná se zejména o vyšší vzdálenosti, vyšší přenosové rychlosti (pro optické kabely teoreticky neomezené – dnes běžně 10Gbps), naprostou odolnost proti rušení, galvanické oddělení (např. budov), apod. Optické kabely jsou ideální pro instalace v prostředí s elektromagnetickým rušením (výrobní haly) a odpadá tím problematika souběhu v trasách se silovými rozvody. Používá se hlavně jako páteřní propojení mezi datovými centry a to jak v rámci objektu, tak i mezi objekty v rámci areálu, města, atd. Nicméně se stále více objevují požadavky na instalaci optického vlákna „až na stůl“ přesně v duchu myšlenky FTTx (fiber to the x) x=desk (stůl), building (budova) atd.

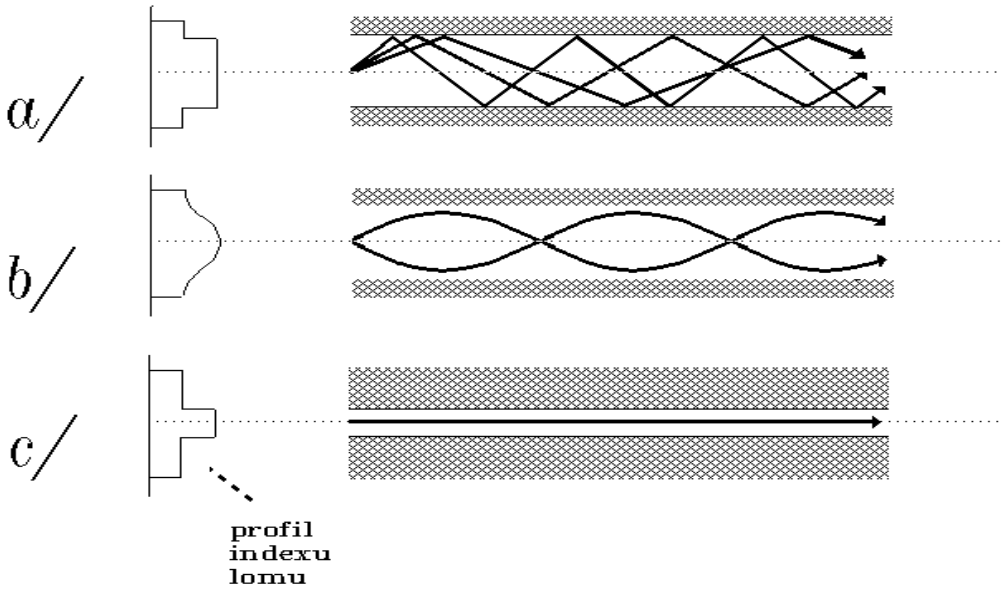
Optické kabely

- Každý kabel obsahuje minimálně 2 optická vlákna (pro každý směr jedno), která jsou obalená sekundární ochranou a plastovým obalem. Data se přenášejí světelnými impulsy. Jeho hlavní výhodou je přenos na velké vzdálenosti. Rozdělujeme je na **mnohovidové optické kabely** a **jednovidové optické kabely.**

- **a) Mnohovidové optické kabely**- původní světelný paprsek je rozložen do více světelných paprsků - dochází k odrazu a lomu od pláště vlákna a následnému zkreslení dat.
- b) **Vylepšená verze mnohovidového kabelu** užívá odrazivou vrstvu s progresivním indexem lomu.
- **c) Jednovidové optické kabely**- původní světelný paprsek prochází jedním optickým vláknem bez lomů a ohybů. Je tedy rychlejší a zaručuje přesnější přenos dat.

Optické kabely

.



Optické kabely

- Vlastní konstrukce kabelu se často velmi liší podle podmínek, pro které je kabel určený. Nějakou konstrukci bude mít optický kabel vedený zavěšením ve vzduchu a jinou konstrukci kabel vedený po mořském dně.
- V základě lze popsat konstrukci takto: Zvenku chrání kabel vrstva polyethylenu, pod ní se často nachází ocelová fólie, zabraňující deformaci kabelu, dále vrstva odpuzující vlhkost.

Poté následuje další vrstva polyethylenu a poté „vrstva“ jednotlivých vláken obalených odrazivou vrstvou. Počet optických vláken se v kabelu různí.

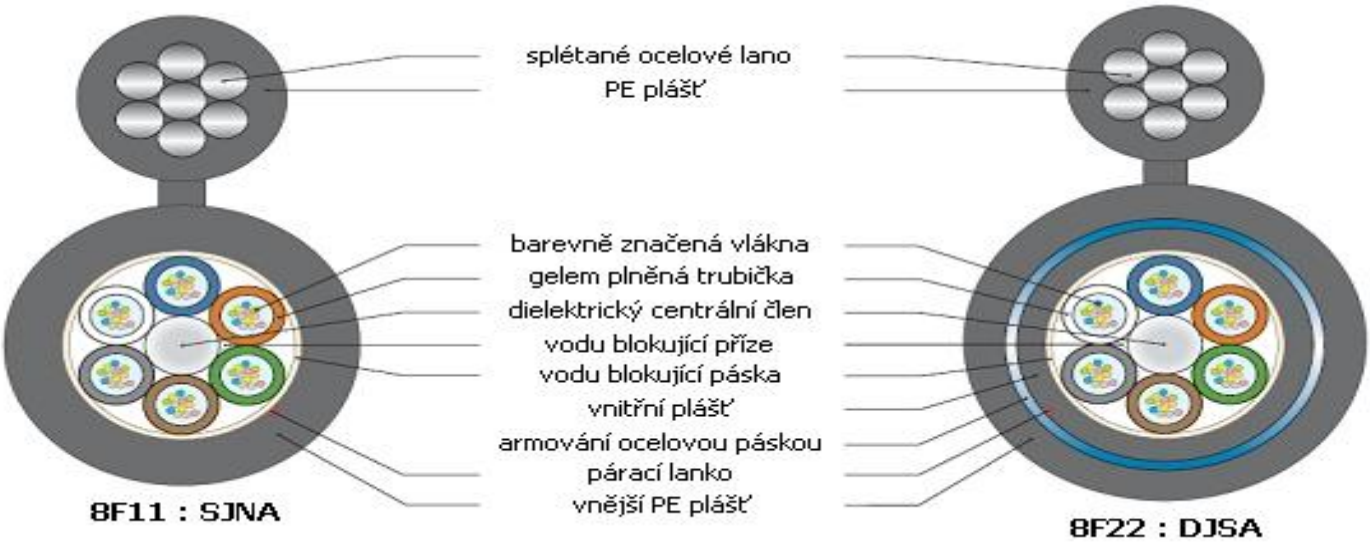
V samotném středu kabelu se nachází dielektrický člen.

- Součástí kabelu mohou bát další vrstvy plnící specifický účel, dále také ocelové lanko pro možnost zavěšení.
- Jako zajímavost lze zmínit kvalitu skla používaného pro optické kabely.

Sklo je tak čisté, že skrz tabuli tohoto skla silnou 10 000 metrů by jste viděli stejně jako přes klasické okno.

Optické Kabely

.



Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány pomocí google.cz

A.3 Síťová architektura

Síťová architektura

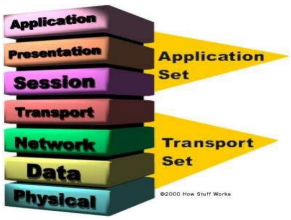
- **Síťová architektura** představuje strukturu řízení komunikace v systémech, tzn. souhrn řídicích činností umožňujících výměnu dat mezi komunikujícími systémy. Komunikace a její řízení je složitý problém, proto se používá rozdělení tohoto problému do několika skupin, tzv. vrstev. Členění do vrstev odpovídá hierarchii činností, které se při řízení komunikace vykonávají.

Vznik

- Počátek síťové architektury se datuje již od vzniku prvních počítačových sítí. Jednalo se však o řešení, které probíhalo v jedné konkrétní firmě, tzv. proprietárně. Brzo se objevily požadavky vytvořit i takovou síť, která by vytvořila jednotný standard a umožnila propojit počítače od různých výrobců, kteří si v té době tvořili své specifické konvence a protokoly.
- Jako první se tohoto úkolu standardizace ujala roku 1977 organizace ISO. Ta vytvořila podkomisi TC97/SC 16, která měla za úkol vytvořit "Architekturu otevřených systémů".Přívlastek "otevřený" (open) zde měl zdůraznit, že systém, vyhovující zamýšlenému standardu, bude připraven pro vzájemné propojení se všemi ostatními systémy na celém světě, které budou vyhovovat témuž standardu.
- O něco později byla koncepce přejmenována na "Architektura vzájemného propojování sítí" (*Open Systems Interconnection Architecture*), která se již nezabývala fungováním počítačů jako takových, ale pouze tím, co se týká jejich vzájemné komunikace. Postupem času tak vznikl finální název "Open Systems Interconnection", zkratkou OSI. Výsledkem tedy byl referenční model ISO/OSI, který pracuje se sedmi vrstvami.

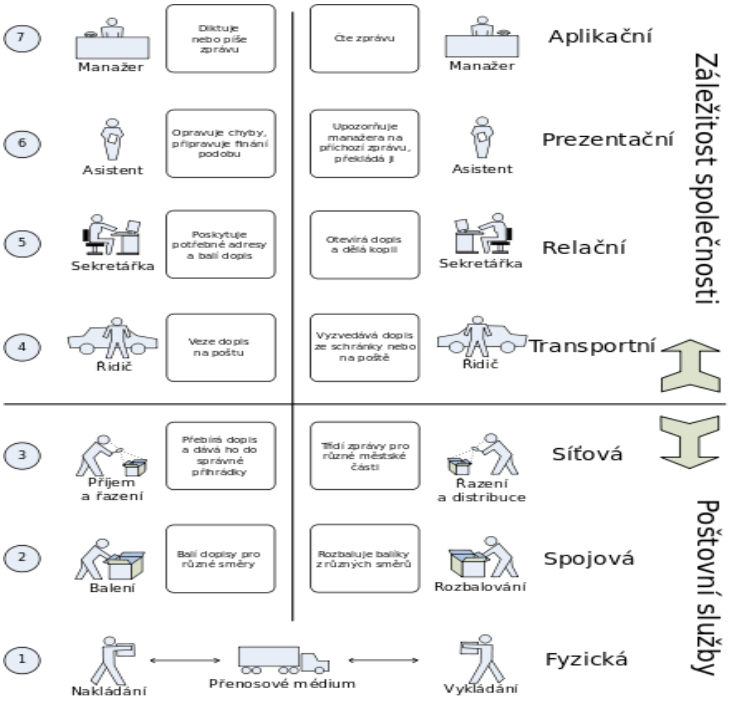
Standartizace

- Cílem standardizace je koncepce umožňující komunikaci nezávisle na technickém provedení (výrobci). Výsledkem je referenční model ISO/OSI. Komunikace je rozčleněna na 7 základních vrstev se specifickými funkcemi a službami. Předpokladem vrstevnatého modelu je, že komunikace probíhá v přesně vymezených přechodových bodech SAP (Service Access Points) a přesně definovaném rozhraní, které vymezuje jednotlivé služby, jejich způsob volání, výpočty parametrů, atd. Komunikace probíhající mezi stejnolehlými vrstvami („peer“) definuje soubor pravidel označovaný jako protokol. Jedna a tatáž vrstva může používat více protokolů např. typ propojení, na způsobu přenosu, optický kabel.



Vrstvy ISO/OSI

1. Fyzická vrstva – zabezpečuje přenos jednotlivých bitů.
2. Linková vrstva – zajišťuje spolehlivé spojení.
3. Síťová vrstva – zajišťuje adresování a směrování paketů.
4. Transportní vrstva – vytváří, rozkládá data na menší části tzv. pakety.
5. Relační vrstva – vytváří časové intervaly pro komunikaci.
6. Prezentační vrstva – určuje a upravuje tvar dat (komprimace).
7. Aplikační vrstva – poskytuje podpůrné funkce konkrétním aplikacím, elektronická pošta.



Paralela mezi RM – OSI a dopisy

Fyzická vrstva

- Vrstva č. 1, anglicky *physical layer*. Specifikuje fyzickou komunikaci. Aktivuje, udržuje a deaktivuje fyzické spoje (např. komutovaný spoj) mezi koncovými systémy. Fyzické spojení může být dvoubodové (sériová linka) nebo mnohobodové (Ethernet).
- Fyzická vrstva definuje všechny elektrické a fyzikální vlastnosti zařízení. Obsahuje rozložení pinů, napěťové úrovně a specifikuje vlastnosti kabelů; stanovuje způsob přenosu "jedniček a nul". Huby, opakovače, síťové adaptéry a hostitelské adaptéry (Host Bus Adapters používané v síťových úložištích SAN) jsou právě zařízení pracující na této vrstvě.
- Hlavní funkce poskytované fyzickou vrstvou jsou:
- Navazování a ukončování spojení s komunikačním médiem.
- Spolupráce na efektivním rozložení všech zdrojů mezi všechny uživatele.
- Modulace neboli konverze digitálních dat na signály používané přenosovým médiem (a zpět) (A/D, D/A převodníky).

Linková (spojová) vrstva

- Vrstva č. 2, anglicky *data link layer*. Poskytuje spojení mezi dvěma sousedními systémy. Uspořádává data z fyzické vrstvy do logických celků známých jako rámce (frames). Seřazuje přenášené rámce, stará se o nastavení parametrů přenosu linky, oznamuje neopravitelné chyby. Formátuje fyzické rámce, opatřuje je fyzickou adresou a poskytuje synchronizaci pro fyzickou vrstvu.
- Datová vrstva poskytuje funkce k přenosu dat mezi jednotlivými síťovými jednotkami a detekuje případně opravuje chyby vzniklé na fyzické vrstvě. Nejlepším příkladem je Ethernet. Na lokálních sítích založených na IEEE 802 a některých na IEEE 802 sítích jako je FDDI, by tato vrstva měla být rozdělena na vrstvu řízení přístupu k médiu (Medium Access Control, MAC) a vrstvu IEEE 802.2 logické řízení linek (Logical Link Control, LLC).
- Na této vrstvě pracují veškeré mosty a přepínače. Poskytuje propojení pouze mezi místně připojenými zařízeními a tak vytváří doménu na druhé vrstvě pro směrové a všesměrové vysílání.

Síťová vrstva

- Vrstva č. 3, anglicky *network layer*. Tato vrstva se stará o směrování v síti a síťové adresování. Poskytuje spojení mezi systémy, které spolu přímo nesousedí. Obsahuje funkce, které umožňují překlenout rozdílné vlastnosti technologií v přenosových sítích.
- Síťová vrstva poskytuje funkce k zajištění přenosu dat různé délky od zdroje k příjemci skrze jednu případně několik vzájemně propojených sítí při zachování kvality služby, kterou požaduje přenosová vrstva. Síťová vrstva poskytuje směrovací funkce a také reportuje o problémech při doručování dat. Veškeré směrovače pracují na této vrstvě a posílají data do jiných sítí. Zde se již pracuje s hierarchickou strukturou adres. Nejznámější protokol pracující na 3. vrstvě je **Internetový Protokol** (IP). Jednotkou informace je paket.

Transportní vrstva

- Vrstva č. 4, anglicky *transport layer*. Tato vrstva zajišťuje přenos dat mezi koncovými uzly. Jejím účelem je poskytnout takovou kvalitu přenosu, jakou požadují vyšší vrstvy. Vrstva nabízí spojově (TCP) a nespojově orientované (UDP) protokoly.
- TCP – Zajišťuje přenos dat se zárukami, který vyžadují aplikace, kde nesmí „chybět ani paket“. Jedná se o přenosy souborů, e-mailů, WWW stránek atd. Záruka se vztahuje na řešení ztrát přenášených paketů, zachování jejich pořadí a odstranění duplikace. Jednotkou posílané informace je na této vrstvě TCP segment.
- UDP – Zajišťuje přenos dat bez záruk, který využívají aplikace, u kterých by bylo na obtíž zdržení (delay) v síti způsobené čekáním na přenos všech paketů a ztráty se dají řešit jiným způsobem (např. snížení kvality, opakování dotazu). Využívá se pro DNS, VoIP, streamované video, internetová rádia, vyhledávání sdílených souborů v rámci sítě DC++, on-line hry atp.

Relační vrstva

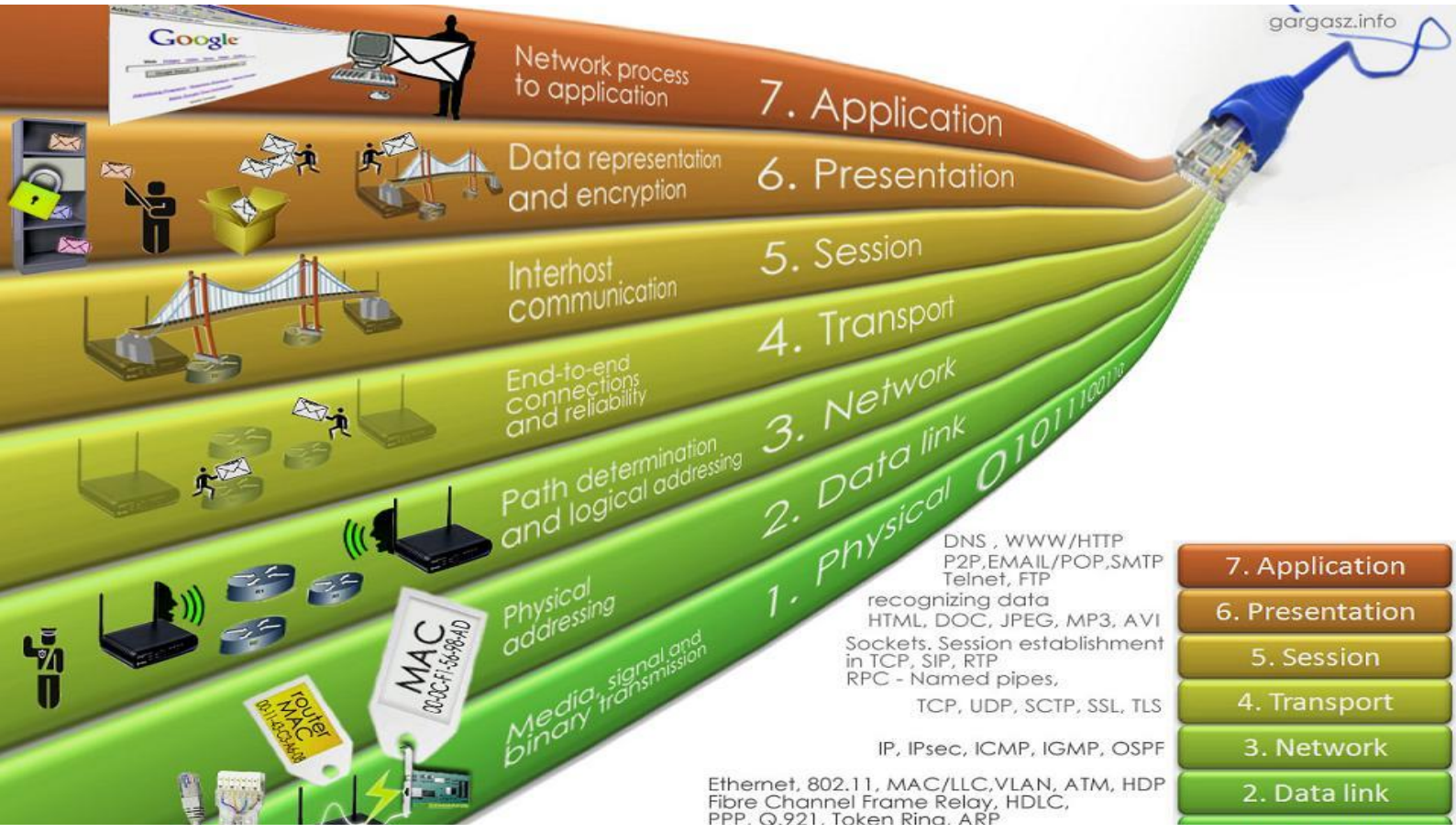
- Vrstva č. 5, anglicky *session layer*. Smyslem vrstvy je organizovat a synchronizovat dialog mezi spolupracujícími relačními vrstvami obou systémů a řídit výměnu dat mezi nimi. Umožňuje vytvoření a ukončení relačního spojení, synchronizaci a obnovení spojení, oznamovaní výjimečných stavů. Do této vrstvy se řadí: NetBIOS, AppleTalk, RPC, SSL. K paketům přiřazuje synchronizační značky, které využije v případě vrácení paketu (např. z důvodu, že se během přenosu dat poškodí síť) k poskládání původního pořadí.

Prezentační vrstva

- Vrstva č. 6, anglicky *presentation layer*. Funkcí vrstvy je transformovat data do tvaru, který používají aplikace (šifrování, konvertování, komprimace). Formát dat (datové struktury) se může lišit na obou komunikujících systémech, navíc dochází k transformaci pro účel přenosu dat nižšími vrstvami. Mezi funkce patří např. převod kódů a abeced, modifikace grafického uspořádání, přizpůsobení pořadí bajtů a pod. Vrstva se zabývá jen strukturou dat, ale ne jejich významem, který je znám jen vrstvě aplikační. Příklady protokolů: SMB(Samba).

Aplikační vrstva

- Vrstva č. 7, anglicky *application layer*. Účelem vrstvy je poskytnout aplikacím přístup ke komunikačnímu systému a umožnit tak jejich spolupráci. Do této vrstvy se řadí například tyto služby a protokoly: FTP, DNS, DHCP, POP3, SMTP, SSH, Telnet, TFTP.



Internetové protokoly na síťových vrstvách

Aplikační vrstva	BitTorrent • DNS • BOOTP • DHCP • FTP • HTTP • HTTPS • IMAP • IRC • Ident • NNTP • NFS • NTP • POP3 • RTP • SIP • SMB • SMTP • SNMP • SSH • STUN • Telnet • Websphere MQ • XMPP
Relační vrstva	SPDY • SSL • NetBIOS • RPC • SMB • NFS
Transportní vrstva	DCCP • IL • RUDP • SCTP • TCP • UDP
Síťová vrstva	IPv4 • IPv6 • ICMP • IGMP • ARP • Proxy ARP • RARP
Linková vrstva	Ethernet • FDDI • PPP • Token ring • Wi-Fi
Fyzická vrstva	10Base2 • 10Base-T • EIA-422 • EIA-485 • RS-232 • RS-449

Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány pomocí google.cz

A.4 Post Office Protocol

Post Office Protocol – POP

(Aplikační vrstva)

- **POP** (*Post Office Protocol*) je internetový protokol, který se používá pro stahování emailových zpráv ze vzdáleného serveru na klienta. Jedná se o aplikační protokol pracující přes TCP/IP připojení. V současnosti je používána zejména třetí verze (POP3), která byla standardizována v roce 1996. POP3 je následníkem protokolů POP1 a POP2 (označení POP už dnes téměř výhradně znamená POP3). V současné době používají téměř všichni uživatelé elektronické pošty pro stahování emailů programy využívající POP3 nebo IMAP.
- Ze vzdáleného serveru se stáhnou všechny zprávy, třeba i ty, které uživatel číst nechce, nebo spam (pokud ho již nefiltruje poštovní server). Většina POP3 serverů sice umožňuje stáhnout i pouze hlavičky zpráv (a následně vybrat zprávy, které se stáhnou celé), ale podpora v klientech vesměs chybí. Tuto nevýhodu může odstranit protokol IMAP, který pracuje se zprávami přímo na serveru.
- Pro odesílání zpráv se používá protokol SMTP, nezávisle na použitém protokolu pro příjem pošty.

Zabezpečení POP3

Jako mnoho jiných starších internetových protokolů, POP3 původně podporoval jenom nešifrované přihlašovací mechanismy. Ačkoli v POP3 je běžný jednoduchý (nezabezpečený) přenos hesel, podporuje současně několik autentizačních metod ověřování na různých úrovních ochrany před neoprávněným přístupem k cizí poštovní schránce. Jedna taková metoda, APOP (kterou základní specifikace definuje jako „volitelný příkaz“), užívá MD5 hash funkci pro zabezpečený přenos hesla od klienta na server. Klienti podporující APOP jsou například Mozilla, Thunderbird, Eudora. Klienti mohou také šifrovat celou POP3 komunikaci užitím SSL nebo modernějšího TLS.

Komunikace

Protokol POP3 má pro své účely vyhrazen TCP port 110. Komunikace probíhá na principu výměny zpráv mezi klientem a serverem. Příkaz vždy začíná na začátku řádky, v základní implementaci POP3 mají příkazy 3 nebo 4 znaky. Příkazy nerozlišují velká a malá písmena. Za příkazem můžou následovat další argumenty, oddělené mezerami. Řádky jsou oddělovány pomocí CRLF. Každá odpověď od serveru musí začínat indikací stavu operace - buď +OK, nebo -ERR. Následovat může textový řetězec s popsaným důvodem stavu. POP3 implementace jsou často poměrně komunikativní a dají se užívat i „ručně“.

Emailoví klienti

- Emailový klient je program, který komunikuje se vzdáleným serverem a pomocí příkazů dokáže uložit emaily na lokální disk a následně je odstranit ze serveru. V současnosti podporuje POP3 většina klientů. Nejběžnější a nejpoužívanější emailoví klienti jsou:
- Microsoft Outlook: asi nejpoužívanější klient, mimo základních funkcí poskytuje také RSS čtečku, kalendář, kontakty, poznámky a možnosti synchronizace s jiným zařízením
- Microsoft Outlook Express: další klient od společnosti Microsoft, který poskytuje pouze základní funkce a na rozdíl od předcházejícího je poskytován s Microsoft Windows zcela zdarma.
- Mozilla Thunderbird: k dispozici jako svobodný software, umožňuje používat mnoho doplňků
- The Bat!: proti předcházejícím klientům obsahuje i integrovaný HTML prohlížeč, možnost ovládání pomocí příkazového řádku a také podporuje doplňky
- Opera: e-mailový klient integrovaný do internetového prohlížeče

Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány pomocí google.cz

A.5 Simple Mail Transfer Protocol

Simple Mail Transfer Protocol – SMTP

(Aplikační vrstva)

Simple Mail Transfer Protocol (zkratka **SMTP**) je internetový protokol určený pro přenos zpráv elektronické pošty (e-mailů) mezi přepravci elektronické pošty (MTA). Protokol zajišťuje doručení pošty pomocí přímého spojení mezi odesílatelem a adresátem; zpráva je doručena do tzv. poštovní schránky adresáta, ke které potom může uživatel kdykoli (off-line) přistupovat (vybírat zprávy) pomocí protokolů POP3 nebo IMAP. Jedná se o jednu z nejstarších aplikací, původní norma RFC 821 byla vydána v roce 1982 (v roce 2001 ji nahradila novější RFC 2821). SMTP funguje nad protokolem TCP, používá port TCP/25.

Architektura pošty

- Doručování elektronické pošty po Internetu se účastní tři druhy programů:
- **MUA** - *Mail User Agent*, poštovní klient, který zpracovává zprávy u uživatele
- **MTA** - *Mail Transfer Agent*, server, který se stará o doručování zprávy na cílový systém adresáta
- **MDA** - *Mail Delivery Agent*, program pro lokální doručování, který umísťuje zprávy do uživatelských schránek, případně je může přímo automaticky zpracovávat (ukládat přílohy, odpovídat, spouštět různé aplikace pro zpracování apod.)

Poštovní klient (MUA)

Poštovní klient je program, který zajišťuje odesílání zpráv a vybírání schránek. Příkladem je např. Microsoft Outlook, Mozilla Thunderbird, Opera, Mutt, Pine a další. Je to v podstatě specializovaný editor, který umí kromě vytvoření zprávy také manipulovat se schránkami, odeslat zprávu nejbližšímu MTA a převzít zprávu ze serveru prostřednictvím POP3 nebo IMAP. Vlastním doručováním zprávy po síti až k adresátovi se klient nezabývá. Součástí klienta bývá také více či méně složitý adresář, který pomáhá uživateli udržet přehled o adresách.

Poštovní server (MTA)

- Poštovní server (MTA) běží obvykle jako démon či Služba Windows a naslouchá na portu TCP/25. K tomuto portu se může připojit (navázat TCP spojení) buď poštovní klient, nebo jiný server, který předá zprávu k doručení. MTA zkontroluje, zda je zpráva určena pro systém, na kterém běží. Pokud ano, předá ji programu MDA (lokální doručení). Pokud je zpráva určena jinému počítači, naváže spojení s příslušným serverem a zprávu mu předá.
- Při vyhledávání vzdáleného serveru, kterému má předat zprávu, musí MTA spolupracovat se systémem DNS. Od serveru DNS si vyžádá tzv. *MX záznam* pro cílovou doménu, který obsahuje IP adresu počítače, který se stará o doručení pošty v této doméně. Pokud DNS tento záznam neobsahuje, pokusí se poštovní server doručit zprávu přímo na počítač uvedený v adrese za zavináčem.
- Poštovní server obsahuje v konfiguraci řadu parametrů, pomocí kterých můžeme mimo jiné nastavit, pro které domény MTA přijímá zprávy. Stejně tak je možné určit, od koho bude nebo nebude zprávy přijímat, což je velmi důležité z hlediska bezpečnosti a ochrany proti spamu.
- Nejčastějšími programy v roli MTA jsou exim, IBM Lotus Domino, Mercury, Microsoft Exchange Server, postfix, qmail, sendmail aj.

Program pro lokální doručování (MDA)

- Server by mohl zprávy do uživatelských schránek ukládat přímo, ale výhodnější je k tomu použít specializovaný program. To umožňuje při doručování ještě dále zprávy zpracovávat nebo filtrovat. Příkladem může být třídění zpráv do různých schránek uživatele podle obsahu (odesilatele, subjektu a pod.), nebo odstraňování nežádoucích zpráv (viry, spam). Tyto volby si může každý uživatel nastavit samostatně nezávisle na ostatních.
- Typickými představiteli MDA jsou procmail a maildrop.

Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány pomocí google.cz

A.6 Internet Message Access Protocol

Internet Message Access Protocol – IMAP

(Aplikační vrstva)

IMAP (*Internet Message Access Protocol*) je internetový protokol pro vzdálený přístup k e-mailové schránce prostřednictvím e-mailového klienta. IMAP nabízí oproti jednodušší alternativě POP3 pokročilé možnosti vzdálené správy (práce se složkami a přesouvání zpráv mezi nimi, prohledávání na straně serveru a podobně) a práci v tzv. on-line i off-line režimu. V současné době se používá protokol **IMAP4** (IMAP version 4 revision 1 - IMAP4rev1), který je definován v RFC 3501.

Možnosti protokolu

Protokol IMAP umožňuje trvalé (tzv. on-line) připojení k e-mailové schránce. Díky tomu je možné s celou poštovní schránkou plně pracovat z libovolného místa. Všechny zprávy a složky jsou uloženy na poštovním serveru a na počítač se stahují jen nezbytné informace, takže při zobrazení složky se stáhnou jen záhlaví zpráv a jejich obsah až v případě, že zprávu chce uživatel přečíst. U jednotlivých zpráv se uchovává jejich stav (nepřečtená, odpovězená, důležitá), uživatel může zprávy přesouvat mezi složkami, složky vytvářet, mazat, prohledávat na straně serveru apod. Protokol umožňuje současné připojení více klientů zároveň. Je také možné zakódovat komunikaci IMAP4 použitím SSL. Buď se komunikuje přes SSL tunel na portu 993, nebo se v komunikaci přes IMAP4 použije STARTTLS (Transport layer security). Protokol IMAP standardně používá port 143 protokolu TCP.

E-mailoví klienti

- E-mailoví klienti jsou obecně konfigurováni buď pro POP3 nebo IMAP4 k přijímání e-mailů a v obou případech používají SMTP pro odesílání. Většina e-mailových programů může také používat Lightweight Directory Access Protokol (LDAP) pro udržování adresářů a práci s informacemi o uživatelích.
- IMAP se mj. často používá v rozlehlých sítích, např. v e-mailovém systému vysokých škol nebo firem. Jakmile jsou nové e-maily uloženy na síti, mohou k nim uživatelé pomocí IMAP hned přistupovat na svých počítačích. S protokolem POP3 uživatelé buď stahují e-maily do svých počítačů nebo k nim přistupují přes webový prohlížeč. Obě tyto metody jsou delší než IMAP a uživatelé musí buď stáhnout všechny nové e-maily, nebo *obnovit* stránku k zobrazení nových zpráv.

Srovnání s POP3

Výhody:

- Hlavní výhody jsou spojené s principem synchronizace, která umožňuje spravování zpráv v emailovém klientu zrcadlit na server, zatímco protokol POP3 je založen především na jednostranném stahování nepřečtených zpráv ze serveru do klienta.
- Stálé připojení: Pokud používáme POP3, klienti se připojí na server pouze na tak dlouho, aby si stáhli novou poštu. Pro použití IMAP4 jsou klienti připojeni tak dlouho, dokud je aktivní uživatelské rozhraní, stahování zpráv je závislé na požadavcích. Pro uživatele s mnoha nebo s velkými e-maily je tento způsob rychlejší.
- Více současně připojených klientů: Protokol POP3 dovoluje připojení pouze jednoho uživatele ke schránce. IMAP dovoluje současné připojení více uživatelů k jedné schránce, a umožňuje vidět změny provedené ostatními klienty.
- Podpora formátu MIME: Téměř všechny e-maily jsou přenášeny ve formátu MIME, což dovoluje zprávám mít stromovou strukturu, kde listové uzly jsou všechny varianty jednotlivých částí obsahu a nelistové uzly jsou varianty více částí. IMAP4 Protokol dovoluje klientům odděleně přijímat jednotlivé MIME části zprávy. Mechanismus umožňuje klientům přijímat textové zprávy, aniž by se zatěžovala linka stahováním přiložených souborů.
- Informace o stavu zprávy: Díky použití příznaků definovaných v protokolu IMAP4 si mohou klienti udržovat přehled o stavu zprávy, např. jestli zpráva byla přečtena, bylo na ni odpovězeno, nebo byla smazána. Tyto příznaky jsou uloženy na serveru, takže různí klienti současně přistupující k jedné schránce v různou dobu mohou zjistit změny provedené ostatními klienty. POP3 něco takového nedovoluje, pokud se uživatel připojí dvěma různými klienty, není možnost tyto informace mezi nimi synchronizovat.
- Webmailové služby jako je Gmail většinou IMAP podporují.
- Práce se složkami na serveru: IMAP4 klienti mohou vytvářet, přejmenovávat anebo mazat mailové schránky (obvykle uváděné uživateli jako složky) na serveru a přenášet zprávy mezi schránkami. Podpora více schránek dovoluje serverům zpřístupnit sdílené a veřejné složky.
- Vyhledání ve zprávách na serveru: IMAP4 poskytuje klientům mechanismus, kterým mohou vyhledávat na serveru zprávy podle různých kritérií. Tento mechanismus dovoluje klientům vyhledávat přímo na serveru, bez nutnosti poštu stáhnout.
- Rozšíření: Na zkušenostech s dřívějšími internetovými protokoly, IMAP4 určuje explicitní mechanismus, podle kterého může být rozšířen. Bylo navrženo mnoho rozšíření základního protokolu, která se běžně používají. IMAP2bis neměl žádný rozšiřující mechanismus a POP3 rozšíření definována v RFC 2449.

Nevýhody:

- Oproti protokolu POP3 je IMAP4 velmi komplikovaný protokol. Jeho implementace je značně složitější a tedy i náchylnější k chybám než implementace POP3. Navzdory tomu IMAP používá mnoho e-mailových serverů a klientů jako jejich standardní přístupovou metodu.
- Pokud nejsou ukládací a vyhledávací algoritmy na serveru bezpečně implementovány, prohledávání velké schránky může značně zatěžovat server. IMAP4 klienti mohou způsobit zpoždění při vytváření nových zpráv, u pomalých připojení (např. u mobilních zařízení). U těchto zařízení je lepší použít Push IMAP, což je rozšířený IMAP protokol o implementaci Push e-mail. Nicméně Push IMAP se běžně nepoužívá a v současnosti IETF pracuje na jiném způsobu.

Emailoví klienti s podporou IMAP

- Alpine - novější nástupce Pine (pod licencí Apache)
- Apple Mail
- Evolution - emailový klient pro Gnome
- KMail - emailový klient pro KDE
- Microsoft Office Outlook, Outlook Express
- Mozilla Seamonkey, Mozilla Thunderbird
- Netscape Mail
- Opera M2
- Pine - multiplatformní klient, průkopník používání protokolu IMAP
- The Bat!

Zdroje informací:

- cs.wikipedia.org

A.7 Domain Name Systém

Domain Name Systém – DNS

(Aplikační vrstva)

- **DNS** (Domain Name System) je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si vyměňují informace. Jeho hlavním úkolem a příčinou vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě. Později ale přibral další funkce (např. pro elektronickou poštu či IP telefonii) a slouží dnes de facto jako distribuovaná databáze síťových informací.
- Protokol používá porty TCP/53 i UDP/53, je definován v RFC1035. Servery DNS jsou organizovány hierarchicky, stejně jako jsou hierarchicky tvořeny názvy domén. Jména domén umožňují lepší orientaci lidem, adresy pro stroje jsou však vyjádřeny pomocí adres 32bitových (IPv4) A záznam nebo 128bitových (IPv6) - AAAA záznam. Systém DNS umožňuje efektivně udržovat decentralizované databáze doménových jmen a jejich překlad na IP adresy. Stejně tak zajišťuje zpětný překlad IP adresy na doménové jméno - PTR záznam.

Jak DNS funguje

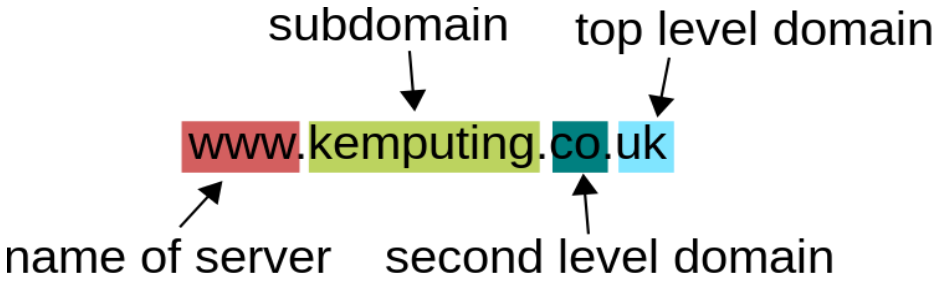
- Prostor doménových jmen tvoří strom s jedním kořenem. Každý uzel tohoto stromu obsahuje informace o části jména (doméně), které je mu přiděleno a odkazy na své podřízené domény. Kořenem stromu je tzv. kořenová doména, která se zapisuje jako samotná tečka. Pod ní se v hierarchii nacházejí tzv. *domény nejvyšší úrovně (Top-Level Domain, TLD)*. Ty jsou buď tematické (*com* pro komerci, *edu* pro vzdělávací instituce atd.) nebo státní (*cz* pro Česko, *sk* pro Slovensko, *ru* pro Rusko atd.).
- Strom lze administrativně rozdělit do zón, které spravují jednotliví správci (organizace nebo i soukromé osoby), přičemž taková zóna obsahuje autoritativní informace o spravovaných doménách. Tyto informace jsou poskytovány autoritativním DNS serverem.
- Výhoda tohoto uspořádání spočívá v možnosti zónu rozdělit a správu její části svěřit někomu dalšímu. Nově vzniklá zóna se tak stane autoritativní pro přidělený jmenný prostor. Právě možnost delegování pravomocí a distribuovaná správa tvoří klíčové vlastnosti DNS a jsou velmi podstatné pro jeho úspěch. Ve vyšších patrech doménové hierarchie platí, že zóna typicky obsahuje jednu doménu. Koncové zóny přidělené organizacím připojeným k Internetu pak někdy obsahují několik domén – například doména *kdesi.cz* a její poddomény *vyroba.kdesi.cz*, *marketing.kdesi.cz* a *obchod.kdesi.cz* mohou být obsaženy v jedné zóně a obhospodařovány stejným serverem.

Složení doménového jména

- Prostor doménových jmen tvoří strom s jedním kořenem. Každý uzel tohoto stromu obsahuje informace o části jména (doméně), které je mu přiděleno a odkazy na své podřízené domény. Kořenem stromu je tzv. kořenová doména, která se zapisuje jako samotná tečka. Pod ní se v hierarchii nacházejí tzv. *domény nejvyšší úrovně (Top-Level Domain, TLD)*. Ty jsou buď tematické (*com* pro komerci, *edu* pro vzdělávací instituce atd.) nebo státní (*cz* pro Česko, *sk* pro Slovensko, *ru* pro Rusko atd.).
- Strom lze administrativně rozdělit do zón, které spravují jednotliví správci (organizace nebo i soukromé osoby), přičemž taková zóna obsahuje autoritativní informace o spravovaných doménách. Tyto informace jsou poskytovány autoritativním DNS serverem.
- Výhoda tohoto uspořádání spočívá v možnosti zónu rozdělit a správu její části svěřit někomu dalšímu. Nově vzniklá zóna se tak stane autoritativní pro přidělený jmenný prostor. Právě možnost delegování pravomocí a distribuovaná správa tvoří klíčové vlastnosti DNS a jsou velmi podstatné pro jeho úspěch. Ve vyšších patrech doménové hierarchie platí, že zóna typicky obsahuje jednu doménu. Koncové zóny přidělené organizacím připojeným k Internetu pak někdy obsahují několik domén – například doména *laura.cz* a její poddomény kosmetika.laura*.cz*, *modeling.laura.cz* mohou být obsaženy v jedné zóně a obhospodařovány stejným serverem.

Složení doménového jména

- Celé jméno se skládá z několika částí oddělených tečkami. Na jeho konci se nacházejí domény nejobecnější, směrem doleva se postupně konkretizuje.
- část nejvíce vpravo je doména nejvyšší úrovně, např. *wikipedia.org* má TLD *org*.
- jednotlivé části (subdomény) mohou mít až 63 znaků a skládat se mohou až do celkové délky doménového jména 255 znaků. Doména může mít až 127 úrovní. Některé implementace jsou však omezeny více.



DNS servery (name servery)

DNS server může hrát vůči doméně (přesněji zóně, ale ve většině případů jsou tyto pojmy zaměnitelné) jednu ze tří rolí:

- **Primární server** je ten, na němž data vznikají. Pokud je třeba provést v doméně změnu, musí se editovat data na jejím primárním serveru. Každá doména má právě jeden primární server.
- **Sekundární server** je automatickou kopií primárního. Průběžně si aktualizuje data a slouží jednak jako záloha pro případ výpadku primárního serveru, jednak pro rozkládání zátěže u frekventovaných domén. Každá doména musí mít alespoň jeden sekundární server.
- **Pomocný (caching only) server** slouží jako vyrovnávací paměť pro snížení zátěže celého systému. Uchovává si odpovědi a poskytuje je při opakování dotazů, dokud nevyprší jejich životnost.

Odpověď pocházející přímo od primárního či sekundárního serveru je autoritativní, čili je brána za správnou. Z hlediska věrohodnosti odpovědí není mezi primárním a sekundárním serverem rozdíl, oba jsou autoritativní. Naproti tomu odpověď poskytnutá z vyrovnávací paměti není autoritativní. Klient může požádat o autoritativní odpověď, v běžných případech ale stačí jakákoli.

Root servery

- Kořenové jmenné servery (*root name servers*) představují zásadní část technické infrastruktury Internetu, na které závisí spolehlivost, správnost a bezpečnost operací na internetu. Tyto servery poskytují kořenový zónový soubor (*root zone file*) ostatním DNS serverům. Jsou součástí DNS, celosvětově distribuované databáze, která slouží k překladu unikátních doménových jmen na ostatní identifikátory.
- Kořenový zónový soubor popisuje, kde se nacházejí autoritativní servery pro domény nejvyšší úrovně. Tento kořenový zónový soubor je relativně velmi malý a často se nemění – operátoři root serverů ho pouze zpřístupňují, samotný soubor je vytvářen a měněn organizací IANA.
- Pojem root server je všeobecně používán pro 13 kořenových jmenných serverů. Root servery se nacházejí ve 34 zemích světa, na více než 80 místech. Root servery jsou spravovány organizacemi, které vybírá IANA. Následující tabulka zobrazuje těchto 13 root serverů:

Seznam root serverů

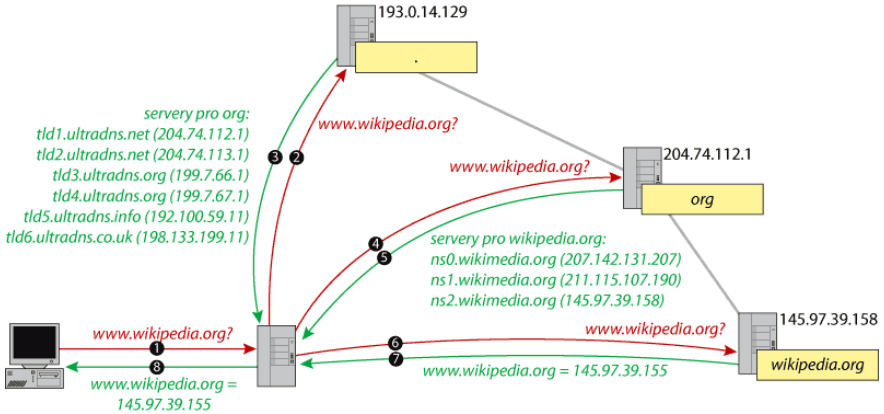
Název root serveru	Operátor
A	VeriSign Global Registry Services
B	University of Southern California - Information Sciences Institute
C	Cogent Communications
D	University of Maryland
E	NASA Ames Research Center
F	Internet Systems Consortium, Inc.
G	U.S. DOD Network Information Center
H	U.S. Army Research Lab
I	Autonomica/NORDUnet
J	VeriSign Global Registry Services
K	RIPE NCC
L	ICANN
M	WIDE Proje

Řešení dotazu

- Každý koncový počítač má ve své konfiguraci síťových parametrů obsaženu i adresu lokálního DNS serveru, na nějž se má obracet s dotazy. V operačních systémech odvozených od Unixu je obsažena v souboru */etc/resolv.conf*, v MS Windows ji najdete ve vlastnostech protokolu TCP/IP (případně můžete z příkazového řádku v XP zadat textový příkaz *ipconfig /all*). Adresu lokálního serveru počítač typicky obdrží prostřednictvím DHCP.
- Pokud počítač hledá určitou informaci v DNS (např. IP adresu k danému jménu), obrátí se s dotazem na tento lokální server. Každý DNS server má ve své konfiguraci uvedeny IP adresy kořenových serverů (autoritativních serverů pro kořenovou doménu). Obrátí se tedy s dotazem na některý z nich.
- Kořenové servery mají autoritativní informace o kořenové doméně. Konkrétně znají všechny existující domény nejvyšší úrovně a jejich autoritativní servery. Dotaz je tedy následně směrován na některý z autoritativních serverů domény nejvyšší úrovně, v níž se nachází cílové jméno. Ten je opět schopen poskytnout informace o své doméně a posunout řešení o jedno patro dolů v doménovém stromě. Tímto způsobem řešení postupuje po jednotlivých patrech doménové hierarchie směrem k cíli, až se dostane k serveru autoritativnímu pro hledané jméno, který pošle definitivní odpověď.
- Získávání informací z takového systému probíhá rekurzí. Resolver (program zajišťující překlad) postupuje od kořene postupně stromem směrem dolů dokud nenalezne autoritativní záznam o hledané doméně. Jednotlivé DNS servery jej postupně odkazují na autoritativní DNS pro jednotlivé části jména.

Příklad řešení dotazu

- Uživatel zadal do svého WWW klienta doménové jméno *www.wikipedia.org*. Resolver v počítači se obrátil na lokální DNS server s dotazem na IP adresu pro *www.wikipedia.org*.
- Lokální DNS server tuto informaci nezná. Má však k dispozici adresy kořenových serverů. Na jeden z nich se obrátí (řekněme na 193.0.14.129) a dotaz mu přepošle.
- Kořenový server také nezná odpověď. Ví však, že existuje doména nejvyšší úrovně *org*, a jaké jsou její autoritativní servery, jejichž adresy tazateli poskytne.
- Lokální server jeden z nich vybere (řekněme, že zvolí *tld1.ultradns.net* s IP adresou 204.74.112.1) a pošle mu dotaz na IP adresu ke jménu *www.wikipedia.org*.
- Oslovený server informaci opět nezná, ale poskytne IP adresy autoritativních serverů pro doménu *wikipedia.org*. Jsou to *ns0.wikimedia.org* (207.142.131.207), *ns1.wikimedia.org* (211.115.107.190) a *ns2.wikimedia.org* (145.97.39.158).
- Lokální server opět jeden z nich vybere a pošle mu dotaz na IP adresu ke jménu *www.wikipedia.org*.
- Jelikož toto jméno se již nachází v doméně *wikipedia.org*, dostane od jejího serveru nepochybně autoritativní odpověď, že hledaná IP adresa zní 145.97.39.155
- Lokální DNS server tuto odpověď předá uživatelskému počítači, který se na ni ptal.



Registrace domény

- Internet Corporation for Assigned Names and Numbers (ICANN) je organizace, která má na starost přidělování a správu doménových jmen a IP adres. Zastřešuje také další regionální organizace, které působí na jednotlivých kontinentech. Každý stát má potom určeného správce zóny, který se stará o příslušnou TLD. Správce buď může domény registrovat sám nebo prostřednictvím tzv. doménových registrátorů, kteří mají náležitá oprávnění.
- ICANN zveřejňuje kompletní seznam všech TLD správců. Informace o vlastnících domén jsou udržovány v online databázi, která je dostupná přes službu WHOIS. Doménové registry obsahují informace o více než 240 národních a generických doménách (ccTLD, gTLD). Např. v ČR se o doménu .cz stará CZ.NIC z.s.p.o..

Příklad založení domény druhého řádu v ČR

- Zvolíme si jméno domény – např. mojedomena.cz
- Zvolíme vhodného registrátora
- Registrátor zřídí primární a sekundární servery
- Požádá správce zóny .cz o registraci domény
- Správce zavede adresy NS serverů do .cz zóny

Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány na **Wikimedia Commons**

A.8 Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol – DHCP

(Aplikační vrstva)

DHCP (anglicky **Dynamic Host Configuration Protocol**) je v informatice název protokolu z rodiny TCP/IP nebo označení odpovídajícího DHCP serveru či klienta. Používá se pro automatickou konfiguraci počítačů připojených do počítačové sítě. DHCP server přiděluje počítačům pomocí DHCP protokolu zejména IP adresu, masku sítě, implicitní bránu a adresu DNS serveru. Platnost přidělených údajů je omezená, proto je na počítači spuštěn DHCP klient, který jejich platnost prodlužuje.

Charakteristika

DHCP protokol umožňuje prostřednictvím DHCP serveru nastavovat stanicím v počítačové síti sadu parametrů nutných pro komunikaci pomocí IP protokolu (tj. využívat rodinu protokolů TCP/IP). Umožňuje předávat i doplňující a uživatelsky definované parametry. Významným způsobem tak zjednodušuje a centralizuje správu počítačové sítě (například při přidávání nových stanic, hromadné změně parametrů nebo pro skrytí technických detailů před uživateli). DHCP servery mohou být sdruženy do skupin, aby bylo přidělování adres odolné vůči výpadkům. Pokud klient některým parametrům nerozumí, ignoruje je.

Typicky se pomocí DHCP nastavují tyto parametry:

- IP adresa
- maska sítě
- implicitní brána (anglicky *default gateway*)
- DNS server (seznam jedné nebo více IP adres DNS serverů)
- a další údaje, např. servery pro NTP, WINS, …

Princip činnosti

- Klienti žádají server o IP adresu, ten u každého klienta eviduje půjčenou IP adresu a čas, do kdy ji klient smí používat (*doba zapůjčení*, anglicky *lease time*). Poté co vyprší, smí server adresu přidělovat jiným klientům.
- Klient komunikuje na UDP portu 68, server naslouchá na UDP portu 67. Po připojení do sítě klient vyšle broadcastem DHCPDISCOVER paket. Na ten odpoví DHCP server paketem DHCPOFFER s nabídkou IP adresy. Klient si z (teoreticky několika) nabídek vybere jednu IP adresu a o tu požádá paketem DHCPREQUEST. Server mu ji vzápětí potvrdí odpovědí DHCPACK. Jakmile klient obdrží DHCPACK, může už IP adresu a zbylá nastavení používat. Klient musí před uplynutím *doby zapůjčení* z DHCPACK obnovit svou IP adresu. Pokud lhůta uplyne aniž by dostal nové potvrzení, klient musí IP adresu přestat používat.
- Protokol definuje roli i tzv. DHCP *relay agenta*. Používá se v situaci, kdy existují dvě nebo více sítí oddělené směrovačem a jen jedna síť obsahuje DHCP server. V takovém případě správce na směrovači zapne relay agenta a nastaví jej tak, aby všesměrové (*broadcast*) DHCP dotazy ze sítí bez DHCP serveru přeposílal DHCP serveru. Agent k přeposílanému dotazu přidá číslo sítě a masku sítě, na které klienta zaslechl, aby DHCP server poznal, ze kterého adresního rozsahu má klientovi adresu přiřadit.

Možnosti přidělení IP adresy

IP adresa může být stanici přidělena několika způsoby:

- **Ruční nastavení**: V tomto případě správce sítě nevyužívá DHCP serveru a konfiguraci jednotlivých stanic zapisuje jednotlivě přímo do konfigurace jednotlivých stanic.
- **Statická alokace**: DHCP server obsahuje seznam MAC adres a k nim příslušným IP adres. Pokud je žádající stanice v seznamu, dostane vždy přidělenu stejnou pevně definovanou IP adresu.
- **Dynamická alokace**: Správce sítě na DHCP serveru vymezí rozsah adres, které budou přidělovány stanicím, které nejsou registrovány. Časové omezení pronájmu IP adresy dovoluje DHCP serveru již nepoužívané adresy přidělovat jiným stanicím. Registrace dříve pronajatých IP adres umožňuje DHCP serveru při příštím pronájmu přidělit stejnou IP adresu.

V IPv6 sítích je automatickému nastavení stanice věnována vyšší pozornost, aby byla konfigurace počítačové sítě ještě jednodušší.

Zpětná kompatibilita

- Protokol DHCP nebyl se svým předchůdcem BOOTP zpětně kompatibilní, což je pro internetové protokoly a vydávání RFC velmi nezvyklé. Protokol DHCP přinesl pouze možnost „pronájmu IP adresy“. Vzhledem k modularitě BOOTP protokolu ale bylo možné tuto vlastnost implementovat i do tohoto předchůdce. V tehdejších počítačových sítích (unixové stanice, DOS s NCSA Telnetem, ve Windows klient Trumpet Winsock) byl protokol BOOTP běžně používán.
- Zpětně nekompatibilní protokol prosadila firma Microsoft, která pro systémy Windows 95 a novější implementovala jako standardní součást pouze podporu protokolu DHCP. Pro správce tak bylo tehdy nutné společně s novou verzí stolního systému Windows nakoupit a provozovat též serverovou edici Windows NT, protože podpora DHCP byla do stávajících BOOTP serverů (typicky provozovaných na unixových systémech) implementována až se zpožděním.

Zdroje informací:

- cs.wikipedia.org
- DHCP Protocol Messages – Popis DHCP na stránkách znalostní databáze firmy Microsoft

A.9 Hypertext Transfer Protocol

Hypertext Transfer Protocol – HTTP

(Aplikační vrstva)

- **HTTP** (Hypertext Transfer Protocol) je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v RFC 2616. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

- V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor (podobně jako e-mail), používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

- HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

Samotný protokol HTTP neumožňuje šifrování ani zabezpečení integrity dat. Pro zabezpečení HTTP se často používá TLS spojení nad TCP. Toto použití je označováno jako HTTPS.

Činnost protokolu

- Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu.

- Pokud uživatel bude mít po chvíli další dotaz na stejný server (např. proto, že uživatel v dokumentu kliknul na hypertextový odkaz), bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, jestli tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká *bezestavový protokol* – protokol neumí uchovávat stav komunikace, dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho „nákupního košíku“ apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

Zabezpečené HTTP

Existují dvě metody zabezpečeného http připojení: HTTPS URI a nadstavba HTTP 1.1 představená v RFC 2817. Druhou metodu ovšem zatím prohlížeče moc nepodporují, takže HTTPS se k vytvoření zabezpečené komunikace používá nejčastěji.

1. **HTTPS URI:** Je syntakticky identické jako http, pouze přidává signalizaci prohlížeči, aby použil šifrovací metodu SSL/TLS k přenosu dat. SSL je vhodné pro HTTP, protože dokáže poskytnout ochranu přenosu, i když je pouze jedna strana komunikace ověřená. Typicky je ověřen pouze server (např. uživatel potvrdí certifikát). Aby pomocí HTTPS bylo možné rozlišovat virtuální servery, existuje rozšíření SNI.

2. **HTTP 1.1 Aktualizovaná hlavička:** HTTP 1.1 představilo podporu pro aktualizaci hlavičky. Klient začíná komunikaci prostým textem, který je později nahrazen TLS. Buď server nebo klient mohou vyžadovat (na požádání), aby byla komunikace převedena na zabezpečenou. Nejběžněji klient začíná prostým textem a to je následováno požadavkem serveru na převod na zabezpečenou komunikaci.

Zdroje informací:

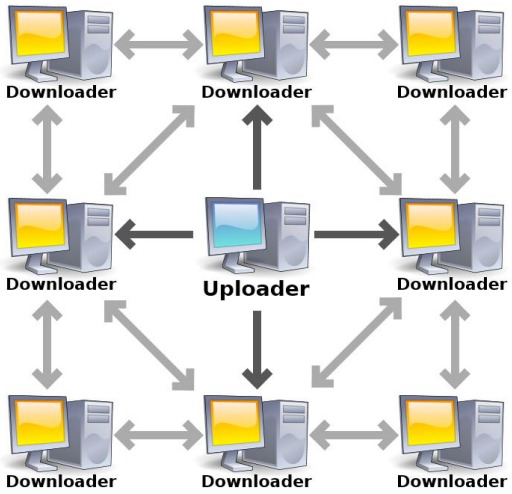
- cs.wikipedia.org
- obrázky vyhledávány na **Wikimedia Commons**

A.10 BitTorrent

BitTorrent
(Aplikační vrstva)

BitTorrent je v informatice nástroj pro peer-to-peer (P2P) distribuci souborů, díky čemuž jsou datové přenosy rozkládány mezi všechny klienty, kteří si data stahují. Velmi populární je při stahování velkých objemů dat (např. distribuce Linuxu, ale většinou warez, viz níže). Název „BitTorrent“ se používá jako název distribučního protokolu, originální klientské aplikace a typu souboru s příponou .torrent.

BitTorrent



Používané pojmy

***Torrent**:* Je soubor .torrent, tedy soubor metadat o sdílených souborech, jejich velikosti a kontrolní součet (viz Hašovací funkce) jednotlivých bloků *torrentu*. Také obsahuje adresu *trackeru* (většinou PHP skript). Dnes je často nahrazován funkcí magnet link. V přeneseném významu pak pod pojmem *torrent* rozumíme i sdílená data.

***Seed**:* Sdílení dat neboli setí. Odvozeně *seeder* je uživatel, který má kompletní kopii *torrentu* a dále sdílí data. Čím více *seederů* je ve *swarmu*, tím větší bývá rychlost downloadu a také se zvyšuje šance na stažení kompletního souboru. Seedováním je torrent udržován v chodu.

***Leech**:* Čili pijavice. Termín *leech* označuje uživatele, který ještě nemá kompletní data torrentu a stále je stahuje.

***Peer**:* Všichni uživatelé, ať už data právě sdílí nebo stahují.

***Uploader**:* Uživatel který vytvořil torrent a umístil jej na tracker.

***Swarm**:* Všichni *peers*, kteří sdílí *torrent*, se nazývají *swarm*. Například šest *leeches* a jeden *seed* je *swarm* (roj) sedmi.

***Aktiv**: Peer* s veřejnou IP nebo s přesměrovaným portem ze serveru poskytovatele připojení. Počítač uživatele je tak "viditelný" z celého internetu a může navazovat spojení s ostatními *peery* ve *swarmu*.

***Pasiv**: Peer* s neveřejnou IP adresou, dostupnou pouze v síti jeho poskytovatele internetu. Pro připojení k internetu používá IP svého poskytovatele. Proto není možné jeho počítač "vidět z venku". Platí tak, že *pasiv* se může spojit pouze s *aktivem*.

***Tracker**:* Je služba, která zprostředkovává spojení mezi klienty (přechovává seznamy IP adres peerů), ale data přes něj netečou, ani nemá žádnou kopii *torrentu*. (Při sdílení musíte nastavit adresu jejího rozhraní pro nabízení torrentů tj nejčastěji „adresa/announce.php“ a následně nahrát .torrent na tracker - to většinou vyžaduje registraci a přihlášení pod vaším jménem).

***Klient**:* Aplikace běžící v počítači uživatele. V její režii probíhá samotné stahování a sdílení dat.

***Announce**:* Komunikace mezi *trackerem* a *klienty* uživatelů.

***DHT**:* Zkratka Distributed Hash Table. Funguje jako distribuovaný *tracker*. Výhodou je, že na rozdíl od normálního trackeru může fungovat i v případě výpadku serveru.

***PEX**:* Zkratka Peer EXchange. Umožňuje *klientům* vyměňovat si mezi sebou seznamy peerů a spojit se tak s uživateli kteří nejsou zahrnuti v seznamu *trackeru*.

***Ratio**:* Poměr mezi odeslanými a staženými daty (odeslaná data/stažená data = ratio). Uživatel by měl dodržovat minimální *ratio* = 1. Obecně se za slušnost považuje alespoň *ratio* = 1,5.

Popis

- Autorem BitTorrentu je Bram Cohen , uvolněn pod licencí *BitTorrent Open Source License* .
- Při distribuci pomocí BitTorrentu jsou soubory (může jich být víc) rozděleny klientem na menší bloky (jejich velikost resp. počet lze nastavit, obvykle mají okolo 250 kB). Každý *leech* může požádat kteréhokoliv *peera* o jemu chybějící blok, a zároveň poskytuje ostatním svoje již kompletně stáhnuté bloky. Často klient může upřednostňovat méně se vyskytující bloky, nebo i bloky na začátcích souborů.

Při obvyklé (klient-server) distribuci souborů klienti stahují data jen ze serveru, který tak musí být výkonný a potřebuje velice rychlé připojení k počítačové síti (Internetu). Protokol BitTorrent umožňuje, aby klient stahoval data nejen ze serveru, ale i od ostatních klientů, kteří mají i jen část potřebných dat. Tím protokol ulehčuje *seedům* (zdrojům s kompletními daty). Stahování přes BitTorrent je tím rychlejší, čím více je *seedů*. Pro velmi populární soubory (obecně se říká, že BT má smysl, pokud v konkrétním torrentu je 10 MB a více, nebo jde o velmi aktuální a populární soubor třeba i s menším objemem dat, ale velkou stahovatelností) může tento torrent žít velmi dlouho. Další aspekt je, že malé soubory se rychle šíří a tvoří velké větve, tj. seedeři mají velké ratio a soubor je velmi rychle rozšířen mezi mnoho klientů, proto může BitTorrent obsloužit tisícenásobně více downloadů než HTTP.

Vytváření a publikování Torrentů

- Peer distribuující data zachází se souborem jako s pevně daným počtem částí s velikostí mezi 64kB a 4MB. Peer vytváří pro každou část kontrolní součet, pomocí SHA algoritmu a zaznamenává je do torrent souboru. Při použití větší velikosti částí souboru se snižuje velikost torrent souboru, což snižuje generovanou zátěž, ale také snižuje efektivitu protokolu. Peer průběžně porovnává předem vypočtené kontrolní součty s daty, která stahuje, a zajišťuje tak integritu dat. Peeři, kteří poskytují kompletní soubory se nazývají seedeři a peeři, kteří poskytují data od začátku se nazývají původní seedeři (initial seeder).
- Struktura dat v torrent souboru závisí na použité verzi BitTorrentu. Podle konvence mají jména torrent souborů příponu .torrent. Torrent soubory mají „oznamovací“ sekci, která specifikuje URL trackeru a takzvanou „info“ sekci, která obsahuje (doporučená) jména souborů, jejich délku, délku částí dat a kontrolní součty všech částí dat. Vše je klientem využíváno k ověření integrity stažených dat.
- Torrent soubory jsou zpravidla zveřejňovány na webových stránkách nebo jinde a jsou spojeny s trackerem. Tracker udržuje seznamy klientů, kteří aktuálně sdílejí torrent. V beztrackerovém systému (decentralizovaný tracking) se všichni peeři chovají jako tracker. Prvním takovým implementovaným klientem byl Azureus, který používal distribuovanou hašovací tabulku (DHT). Později byl Bittorentem vyvinut a adoptován alternativní a nekompatibilní Mainline DHT.
- Pro vytvoření torrentu je třeba uvést také seznam trackerů. Je možné použít veřejné trackery, které shromažďují pouze seznam peerů identifikovaných podle IP adresy a hashe souboru, o který mají zájem. Příklad veřejných trackerů, které můžete použít při vytváření torrentů:
- udp://tracker.publicbt.com:80/announce
- udp://tracker.openbittorrent.com:80/announce
- http://opentracker.nodex.cz:80/announce
- udp://tracker.ccc.de:80/announce

Stahování torrentů a sdílení souborů

- Uživatelé prohledávají web s cílem najít torrent, o který se zajímají, mohou si jej stáhnout a otevřít například BitTorentovým klientem. Klient se spojí s trackerem, který je specifikovaný v torrent souboru. Tracker poskytne klientovi seznam peerů, kteří aktuálně přenášejí části dat daných souborů specifikovaných v torrent souboru. Klient se připojí k těmto peerům a obdrží náhodné části dat. Jestliže swarm obsahuje jenom jednoho seedera, který tato data sdílí od začátku (initial seder), klient se přímo k němu připojí a požaduje části dat.
- Klienti obsahují mechanismus pro optimalizaci jejich stahovacího (download) a nahrávacího (upload) poměru. Například klient záměrně stahuje části dat v náhodném pořadí, aby se zvýšila šance P2P výměny dat, protože je možné jen tehdy, pokud mají peeři staženy různé části dat.
- Efektivita výměny dat závisí hodně na rozhodnutí, komu budou klienti posílat data. Klient může upřednostňovat posílání dat peerum, kteří data zpětně sdílí, což podporuje férovou výměnu. Avšak přísná pravidla často vyústí k neoptimálním situacím, když nově připojení peeři nejsou schopni odesílat data nebo když dva peeři s dobrým vzájemným spojením si data nevyměňují, protože žádný z nich nepřevezme iniciativu. Proti těmto jevům používá oficiální BitTorrent klient mechanismus zvaný „optimistic unchoking“ (optimistické uvolňování), kdy klient rezervuje část přenosové kapacity pro seedování dat náhodným peerům v naději, že najde lepší partnery, a zároveň umožňuje nově příchozím peerům připojení do swarmu.

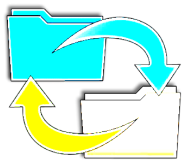
Zdroje informací:

www.root.cz/clanky/bittorrent-technologie/ - Radim Kolář
www.bittorrent.com/
cs.wikipedia.org
obrázky vyhledávány na Wikimedia Commons

A.11 File Transfer Protocom

File Transfer Protocol

(Aplikační vrstva)



FTP (anglicky **File Transfer Protocol**) je v informatice protokol pro přenos souborů mezi počítači pomocí počítačové sítě. Využívá protokol TCP z rodiny TCP/IP a může být používán nezávisle na použitém operačním systému (je platformě nezávislý). Definován byl v roce 1985 v RFC 959. RFC 2228 (červen 1998) navrhlo některá bezpečnostní rozšíření a RFC 2428 (září 1998) přidává podporu pro IPv6 a definuje také novou možnost pasivního režimu. Jeho podpora je součástí webových prohlížečů nebo specializovaných programů (tzv. *FTP klientů*).

Popis

- FTP je jeden z nejstarších protokolů, využívá porty TCP/21 a TCP/20. Port 21 slouží k řízení a jsou jím také přenášeny příkazy FTP. Port 20 slouží k vlastnímu přenosu dat, který je 8bitový. Přenos může být *binární* nebo *ascii* (textový). Při textovém přenosu dochází ke konverzi konců řádků – CR/LF (DOS, Microsoft Windows) nebo jen LF (unixové systémy), pokud jsou koncové systémy rozdílné. Při binárním přenosu není do dat nijak zasahováno.
- Protokol je interaktivní a umožňuje řízení přístupu (přihlašování login/heslo), specifikaci formátu přenášeného souboru (znakově – binárně), výpis vzdáleného adresáře atd. V současné době už není považován za bezpečný a z tohoto důvodu pro něj byla definována některá rozšíření (RFC 2228).
- V protokolu je použit model klient-server. FTP server poskytuje data pro ostatní počítače. Klient se k serveru připojí a může provádět různé operace (výpis adresáře, změna adresáře, přenos dat atd.). Operace jsou řízeny sadou příkazů, které jsou definovány v rámci FTP protokolu, proto kdokoliv může vytvořit klienta pro jakékoliv prostředí nebo operační systém. Existuje mnoho programů pro FTP servery i klienty a mnoho je jich volně dostupných.

Činnost

FTP běžně pracuje na dvou portech, 21 a 20 a běží výhradně přes TCP (Transmission control protocol). FTP server naslouchá na portu 21 na příchozí spojení z FTP klienta. Na tomto portu běží příkazy, které zachytává server. Na portu 20 se přenáší pouze data, nikoliv příkazy. Jakmile se začnou stahovat data, na příkazovém portu se nic nepřenáší. Při stahování velkých souborů přes připojení s firewallem může tento kvůli dlouhodobé nečinnosti zablokovat komunikaci na portu 21.

Aktivní FTP spojení (jak se do lesa volá, tak se z něj ozývá)

- Aktivní spojení je prapůvodní verzí. Funguje tak, že klient vystaví spojení ze svého portu nad 1024/TCP na port 21/TCP FTP serveru (viz obrázek na další stránce).
- Zde probíhá tzv. řídicí spojení, pomocí něhož klient ovládá FTP server. Poté, co se klient autentizuje, může zadávat serveru příkazy, např. LIST/NLIST pro výpis aktuálního adresáře, CWD (Change Working Directory) pro změnu adresáře, STOR pro uložení souboru na server, RETR pro stažení souboru. Tyto příkazy serveru se provádějí pomocí lokálních příkazů poplatných FTP klientovi (unixový/DOSový používá ls, cd, get, put, windowsový zase tlačítka GUI). Server tyto příkazy plní a jejich výstup (ať už výpis adresáře či samotná up/down-loadovaná data) vysílá ze svého portu 20/TCP na port nad 1024/TCP náhodně (klientem) vybraného portu klienta.
- **Jaké jsou výhody?** Textové řídicí spojení je abstrahováno od spojení datového, tudíž se snižuje režie spojení (není třeba složitě odlišovat řízení a data) a hlavně je možno odesílat data úplně jinam, než je klient (představte si – pomocí „superpomalého“ v.90 modemu přenášíte data mezi dvěma servery propojenými páteřní gigabitovou optikou, a to skoro plnou rychlostí onoho gigabitu, ale pozor, tuhle funkci mají dnešní servery kvůli bezpečnosti standardně vypnutou!). Dále takto nakonfigurovaný server můžete velmi jednoduše umístit za NATem – jediné, co musíte zajistit, je portforwarding portu 21/TCP a korektní NAT.
- **Jaké jsou nevýhody?** Největší nevýhoda je ta, že klient musí mít povoleno příchozí spojení z portu 20/TCP na předem neurčený cílový port. To předpokládá, že klienti za NATem budou mít speciálně přizpůsobený router (viz následující zábavná kapitola „Caveats aneb Hnusné triky“).

Pasivní spojení (čili nahraďme jeden problém druhým)

- U pasivního spojení se (s ohledem na klienty za NATem) obě spojení vystavují směrem k FTP serveru (viz. Obrázek na další stránce).
- Přepnutí na tento režim se provede příkazem PASV. Nastává ovšem jiný problém: jak odlišit jednotlivé datové porty jednotlivých klientů? Jistě byste nechtěli stahovat data, která chtěl stahovat jiný uživatel a obráceně – jistě byste nechtěli, aby vaše data stahoval někdo jiný jen proto, že otevřel spojení na datový port serveru dříve než jste to stihli vy. Proto FTP server otevírá pro každý požadavek na datové spojení dedikovaný dočasný server na náhodném portu z rozsahu portů nad 1024/TCP (tento rozsah jde u slušných serverů nastavit). Otázkou je, jak to dá vědět klientovi. Vývojáři se rozhodli, že co nejjednodušeji, tj. v řídicím spojení jako textovou odezvu na příkaz, např.:

➢ PASV

➢ 227 Entering Passive Mode (123,213,231,123,234,100).

➢ LIST

➢ 150 Opening ASCII mode data connection for file list

➢ Stahuje se

- Direktiva z příkladu serveru č. 227 (odpověď na PASV) říká klientovi, že má vystavit následující nové FTP-DATA spojení na předem určený port. V závorce je uvedena adresa a port – první 4 oktety (oktet = číslo oddělené čárkou) odpovídají IP adrese FTP serveru, a poslední dva označují port. Port se spočte jako 5. oktet krát 256 + 6. oktet. V našem případě to je port 60004/TCP. Na tomto portu na nás budou čekat data s výpisem adresáře (viz příklad, ve kterém k nám takto doputují data ze serveru jako odpověď na příkaz LIST).
- **Jaké jsou výhody?** Klient nemusí mít forwardovaný žádný port, pokud je za NATem, a tedy na klienta nejsou činěny žádné nadstandardní požadavky ohledně spojení.
- **Jaké jsou nevýhody?** Server nemůže být za NATem, resp. byste museli zajistit forwarding odpovídající sady vysokých portů a provést další úpravy na routeru.

Nástroje pro připojení na FTP server

Připojení lze provést buď pomocí klienta, který je součástí operačního systému, nebo si stáhnout klienta FTP, nebo využít program s implementovaným FTP klientem jako např. Total Commander.

Použití FTP

Nejčastější užití ftp přenosů jsou:

1. Sdílení dat (často hudba, videa, vlastní tvorba, …).
2. Správa účtů internetových stránek.

Výhody a nevýhody

1. hesla a soubory jsou ve standardním protokolu zasílána jako běžný text (nejsou šifrovaná)
   - snižuje bezpečnost (ohrožuje jméno, heslo, ale i přenášená data)
   - existují rozšíření FTP protokolu, která tento nedostatek odstraňují
2. používají se 2 TCP spojení (první TCP spojení je řídící, druhé datové pro vlastní přenos dat)
   - je-li použit firewall, protokol vyžaduje jeho speciální podporu (aktivní FTP přenos)
   - podpora aktivního přenosu nefunguje u šifrovaného řídícího spojení
   - pasivní přenos tento nedostatek odstraňuje
3. FTP server má delší odezvy
   - nemožnost sloučit přenos více (malých) souborů do jednoho zvyšuje časovou režii i zátěž serveru
   - serverová část je jednodušší, než běžný HTTP server (neplatí pro odlehčené HTTP servery)
   - na rozdíl od HTTP má protokol širší možnosti (nastavení práv, mazání, upload, rekurzivita, FXP, …)
4. v některých sítích je povolen pouze protokol HTTP
   - FTP je v současné době méně používáno

Aktivní a pasivní připojení

Připojení k FTP serveru je možné realizovat v aktivním nebo pasivním režimu. Pasivní režim je bezpečnější, ale ne vždy je technicky realizovatelný.

- **Aktivní režim**: Na portu TCP/20 jsou přenášena data (data connection). V aktivním režimu navazuje připojení pro přenos dat server, klient naslouchá. Problém zpravidla nastává v případě, kdy se klient připojuje z privátní sítě a jeho IP adresa je překládána (NAT).
- **Pasivní režim**: V pasivním režimu navazuje data connection klient, kterému při sestavování připojení poslal server svou IP adresu a TCP port, na kterém naslouchá.
- **FTP server, port forward (PF) a pasivní připojení**: Pokud je připojení k FTP serveru realizováno prostřednictvím PF (nejčastěji se jedná o router s NAT a PF), tak router musí mít následující vlastnost – čte datovou část paketů FTP připojení, zjistí na jakém portu server naslouchá pro navázání data connection klientem a tento port začne forwardovat směrem k serveru. Po ukončení relace je ukončen i popsaný PF. Routery mají zpravidla tuto vlastnost již vestavěnou. V případě systému Linux je nutné na routeru spustit příslušný modul – příkaz je 'modprobe nf_conntrack_ftp'

Bezpečnější použití FTP

- FTP přes SSH označuje tunelování FTP skrz spojení navázaného pomocí SSH protokolu. Někdy je též označováno jako *Bezpečné FTP* (Secure FTP), které by nemělo být zaměňováno za FTPS (tj. FTP s podporou SSL/TLS). Dalšími metodami bezpečného přenosu dat jsou SFTP a SCP, které využívají protokol SSH.
- Protože protokol FTP používá dvě spojení, je velmi těžké zajistit, aby bylo tunelováno nejen řídící, ale i datové spojení. Pokud je FTP klient nastaven do pasivního režimu a instruován pro spojení se SOCKS serverem, které může SSH klient zajistit, může být dosaženo tunelování obou spojení.
- Druhou možností je, aby SSH klient zasahoval do řídícího spojení podobně, jak je to nutné při průchodu FTP skrz NAT. Tuto funkci poskytuje 3. verze *SSH Communications Security's software suite* a program FONC distribuovaný pod GPL licencí.

Zdroje informací:

http://pc.poradna.net/a/view/307878-jak-funguje-ftp

cs.wikipedia.org

obrázky vyhledávány na Wikimedia Commons

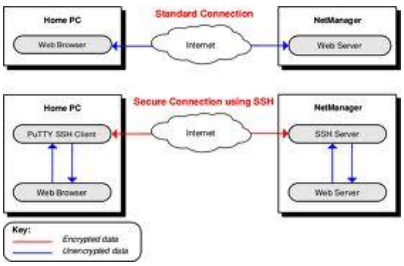
A.12 Secure Shell

Secure Shell – SSH

(Aplikační vrstva)

SSH (**Secure Shell**) je v informatice označení pro program a zároveň pro zabezpečený komunikační protokol v počítačových sítích, které používají TCP/IP. SSH byl navržen jako náhrada za telnet a další nezabezpečené vzdálené shelly (rlogin, rsh apod.), které posílají heslo v nezabezpečené formě a umožňují tak jeho odposlechnutí při přenosu pomocí počítačové sítě. Šifrování přenášených dat, které SSH poskytuje, slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť, jako je například Internet.

Stručný popis



- SSH umožňuje bezpečnou komunikaci mezi dvěma počítači, která se využívá pro zprostředkování přístupu k příkazovému řádku, kopírování souborů a též jakýkoliv obecný přenos dat (s využitím síťového tunelování). Zabezpečuje autentizaci obou účastníků komunikace, transparentní šifrování přenášených dat, zajištění jejich integrity a volitelnou bezeztrátovou kompresi. Server standardně naslouchá na portu TCP/22.
- Označení „Secure Shell“ je mírně zavádějící, protože nejde ve skutečnosti o náhradu shellu ve smyslu interpret příkazů. Název byl odvozen z existujícího programu rsh, který má podobné funkce, ale není zabezpečený.

Použití

SSH je používáno jako bezpečná náhrada starších protokolů a nabízí i nové vlastnosti:

- náhrada protokolu Telnet, práce na vzdáleném počítači přes nezabezpečenou síť
- náhrada protokolu Rlogin, přihlášení na vzdálený počítač
- náhrada protokolu Rsh, spouštění příkazů na vzdáleném počítači
- tunelování spojení
- přesměrování TCP portů a X11 spojení zabezpečeným kanálem
- bezpečný přenos souborů pomocí SFTP nebo SCP
- automatické vzdálené monitorování a management serverů
- bezpečné připojování složek na vzdáleném serveru jako souborový systém na lokálním počítači použitím SSHFS
- prohlížení webu přes šifrované proxy spojení s SSH klientem, který podporuje SOCKS protokol
- plnohodnotnou šifrovanou VPN (pouze OpenSSH server a klient, kteří tuto vlastnost podporují)

Architektura

Protokol SSH-2 má dobře navrženou vnitřní architekturu (RFC 4251) rozdělenou na oddělené vrstvy. Otevřená architektura nabízí významnou flexibilitu umožňující použití SSH nejen pro zabezpečený shell. Funkce transportní vrstvy samotné je srovnatelná s TLS (Transport Layer Security). Vrstva autentizace uživatele je navržena pro snadné rozšíření vlastními autentizačními metodami. Vrstva spojení nabízí použití více podružných relací přenášených jedním SSH spojením, které je srovnatelné s BEEP (Block Extensible Exchange Protocol) a kteroužto vlastnost TLS nenabízí.

Transportní vrstva

Transportní vrstva (RFC 4253) zajišťuje počáteční výměnu klíčů, serverovou autentizaci, kompresi a ověření integrity. Poskytuje vyšší vrstvě prostředí pro posílání a přijímání nešifrovaných až 32,768 bytů dlouhých paketů dat (prostý text, delší mohou být povoleny implementací). Transportní vrstva také zajišťuje opětovnou výměnu klíčů – obvykle po 1 GB přenesených dat nebo po uplynutí 1 hodiny, podle toho, co nastane dříve.

Vrstva autentizace uživatele

- Vrstva autentizace uživatele (RFC 4252) zajišťuje autentizaci klientů, která může být provedena mnoha způsoby. Samotná autentizace je řízena SSH klientem, server pouze reaguje na autorizační požadavky od SSH klienta. Do nejpoužívanějších metod autentizace patří metody:
- Password: Password je metoda pro přímou výměnu hesla, zahrnující možnost změny hesla. Tato metoda není implementována ve všech programech.
- Publickey: Publickey je metoda přihlášení pomocí veřejného klíče, obvykle podporuje alespoň DSA nebo RSA klíče, může podporovat i jiné metody založené na X.509 certifikátech. Tato metoda může zabránit útokům hrubou silou (anglicky brute force), ale pouze tehdy, když je vypnuta metoda "password".
- Keyboard-interactive: Keyboard-interactive je univerzální metoda (RFC 4256), při které server pošle klientovi jednu nebo více výzev, na které uživatel odpovídá pomocí klávesnice. Nejčastěji se používá pro jednorázovou autentizaci jako je S/Key nebo SecurID.
- GSSAPI: GSSAPI metody poskytují rozšiřitelné rozhraní pro autentizaci pomocí externích mechanizmů jako je Kerberos 5 nebo NTLM, poskytujících možnost centrální autentizace (*Single Sign-On*, SSO) pro přihlášení pomocí SSH relace. Tyto metody jsou obvykle používány v komerčních SSH implementacích, určených pro organizace, i když OpenSSH funkční GSSAPI také obsahuje.

Vrstva spojení

Vrstva spojení (RFC 4254) definuje koncept kanálů, požadavků kanálů a globálních požadavků skrze které jsou poskytovány SSH služby. Jedno SSH spojení může hostovat více kanálů zároveň, kdy každý může přenášet data v obou směrech. Požadavky kanálů jsou použity pro přenos mimopásmových dat, jako jsou např. změna velikosti terminálového okna nebo návratový kód procesu na straně serveru. SSH klient si může pomocí globálního požadavku vyžádat forwardování (tunelování) portu na straně serveru. Standardní typy kanálů zahrnují:

- **Shell**: Shell pro terminálové shelly, SFTP a požadavky exec (zahrnující SCP přenosy)
- **Direct-tcpip**: Direct-tcpip pro forwardovaná klient-server spojení
- **Forwarded-tcpip**: Forwarded-tcpip pro forwardovaná server-klient spojení

SSHFP záznamy v DNS

SSHFP záznamy v DNS (RFC 4255) poskytují veřejné otisky klíčů, které usnadňují ověření autenticity hosta (tj. protistrany).

Bezpečnostní výstrahy

- Protokol SSH-1 byl označen za zastaralý kvůli bezpečnostním nedostatkům (např. možnost útoku man-in-the-middle), a proto by jeho použití by mělo být explicitně znemožněno. Současná situace je taková, že většina moderních serverů a klientů používá SSH-2, ale stále existují některé organizace, které používají software bez podpory SSH-2 a tudíž není možné podporu SSH-1 úplně odstranit.
- U všech verzí protokolu SSH je velmi důležité, aby neznámý veřejný klíč byl před jeho schválením řádně ověřen, jinak může dojít k dešifrování důvěrných informací a útokům typu man-in-the-middle.
- Stejně jako každý šifrovaný protokol, může být SSH považováno za bezpečnostní riziko pro firmy nebo vlády, které nevěří svým zaměstnancům a chtějí mít jejich komunikaci pod kontrolou. Navíc má SSH v sobě zabudované jednoduché mechanismy pro vytváření tunelovaných spojení, skrze které lze přenášet velké objemy dat a vytvářet tak nežádoucí vstupní body, které mohou sloužit k úniku důležitých informací nebo k průniku do vnitřní sítě. Stejně tak mohou být tytéž vlastnosti užitečné (např. šifrování služeb jako je POP3 nebo IMAP prostým použitím SSH tunelu), protože je u jiných protokolů nenajdeme.
- Vzhledem k mnoha vlastnostem, které protokol SSH nabízí, se povolení průchodu SSH přes firewall může stát vážným bezpečnostním rizikem. Kromě přesměrováním portů totiž některé implementace SSH přímo podporují Layer2 VPN, což umožňuje efektivní spojení dvou vzdálených ethernetových sítí, jako by byly připojeny ke stejnému switchi. V současnosti se hledá řešení těchto problémů.

Autentizace pomocí veřejného klíče

Pro autentizaci uživatele je možné v SSH použít veřejný klíč. Nejprve je vygenerován pár šifrovacích klíčů – privátní (soukromý) klíč a veřejný klíč. Privátní je bezpečně uložen u uživatele a je chráněn heslovou frází. Veřejný klíč je uložen na cílový server (typicky do domácího adresáře uživatele, v unixových systémech do souboru ~/.ssh/authorized_keys). Při pokusu o přihlášení server veřejným klíčem, který má k dispozici, zašifruje blok náhodných dat (tzv. výzvu typu challenge-response), kterou nelze snadno odvodit nebo uhádnout a pošle ji klientovi. Klient výzvu pomocí privátního klíče dešifruje a dešifrovanou ji pošle zpět serveru. Pokud je výzva správně rozšifrována, má tím server ověřeno, že klient má k dispozici privátní klíč, který odpovídá veřejnému klíči, který má server dispozici, a tudíž může přístup schválit (autorizovat). Pokud výzva nebude správně rozšifrována, bude přístup pro klienta zamítnut. Z toho plyne, že privátní klíč neopustí klientův počítač, tudíž se nemůže stát, že by ho někdo odcizil při přenosu po síti, a přesto může dojít k ověření autenticity klienta a umožnění přístupu.

Seznam implementací

- Lsh, implementace SSH klientu i serveru spravované v projektu GNU
- OpenSSH, open source implementace SSH. Původně odštěpená z originálního SSH-1 (klient i server)
- PuTTY, SSH klient pro Windows
- SSH Tectia Client
- PenguiNet
- SSHDOS
- WinSCP, souborový manažer založený na knihovnách Putty, umožňující práci v režimu sftp a scp
- JavaSSH
- Dropbear, malý klient a server určený pro OS splňující normu POSIX
- Idokorro Mobile SSH, implementace SSH pro RIM BlackBerry a mobilní telefony
- Ganymed-SSH2, Open Source klientská knihovna funkcí SSH2 v jazyku Java
- SSH přes HTTP

Závěr

SSH program je dnes běžně používán při vzdálené práci a pro vzdálenou správu. Klient se při navázání spojení připojuje k SSH démonu (SSH daemon, sshd). SSH démon podle svého nastavení rozhoduje, zda spojení přijme, jakou formu autentizace bude požadovat, případně na kterém portu bude naslouchat. Implementace SSH klientů i serverů (SSH démon) je dostupná téměř pro jakoukoliv platformu. Většinou jsou dostupné jak komerční, tak i Open Source varianty.

Zdroje informací:

www.serverwatch.com/news/print.php/3551081

cs.wikipedia.org

obrázky vyhledávány na Wikimedia Commons

obrázky vyhledávány na Google - obrázky

A.13 Secure Sockets Layer

Secure Sockets Layer – SSL

(Relační vrstva)

Secure Sockets Layer, **SSL** (doslova *vrstva bezpečných socketů*) je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci komunikujících stran. Následovníkem SSL je protokol Transport Layer Security (TLS).

Využití

Protokol SSL se nejčastěji využívá pro bezpečnou komunikaci s webovými servery pomocí HTTPS, což je zabezpečená verze protokolu HTTP. Po vytvoření SSL spojení (*session*) je komunikace mezi serverem a klientem šifrovaná, a tedy zabezpečená.

Obvyklá využití SSL certifikátů:

- on-line obchody, které přijímají objednávky a údaje platebních karet
- www portály a projekty s administrací pro zabezpečení hesel a dat
- komunikace s obchodním partnerem (výměna důvěrných informací)
- zabezpečení přístupu k poště mimo firemní síť (Exchange, ...)
- zpracování citlivých osobních údajů
- dodržení regulačních ustanovení (legislativa), která vyžadují zabezpečené přenosy

Princip

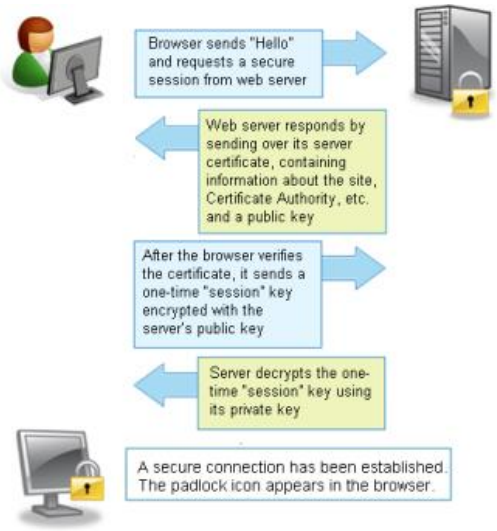
Ustavení SSL spojení funguje na principu asymetrické šifry, kdy každá z komunikujících stran má dvojici šifrovacích klíčů – veřejný a soukromý. Veřejný klíč je možné zveřejnit a pokud tímto klíčem kdokoliv zašifruje nějakou zprávu, je zajištěno, že ji bude moci rozšifrovat jen majitel použitého veřejného klíče svým soukromým klíčem.

Ustavení SSL spojení (*SSL handshake*, tedy „potřásání rukou“) pak probíhá následovně:

- Klient pošle serveru požadavek na SSL spojení, spolu s různými doplňujícími informacemi (verze SSL, nastavení šifrování atd.).
- Server pošle klientovi odpověď na jeho požadavek, která obsahuje stejný typ informací a hlavně certifikát serveru.
- Podle přijatého certifikátu si klient ověří autentičnost serveru. Certifikát také obsahuje veřejný klíč serveru.
- Na základě dosud obdržených informací vygeneruje klient základ šifrovacího klíče, kterým se bude šifrovat následná komunikace. Ten zašifruje veřejným klíčem serveru a pošle mu ho.
- Server použije svůj soukromý klíč k rozšifrování základu šifrovacího klíče. Z tohoto základu vygenerují jak server, tak klient hlavní šifrovací klíč.
- Klient a server si navzájem potvrdí, že od teď bude jejich komunikace šifrovaná tímto klíčem. Fáze handshake tímto končí.
- Je ustaveno zabezpečené spojení šifrované vygenerovaným šifrovacím klíčem.
- Aplikace od teď dál komunikují přes šifrované spojení. Například POST požadavek na server se do této doby neodešle.

Během první fáze ustanovení bezpečného spojení si klient a server dohodnou kryptografické algoritmy, které budou použity. V dnešní implementaci jsou následující volby:

- pro výměnu klíčů: RSA, Diffie-Hellman, DSA nebo Fortezza;
- pro symetrickou šifru: RC2, RC4, IDEA, DES, 3DES nebo AES;
- pro jednocestné hašovací funkce: MD5 nebo SHA.



Certifikační autority

- CA jsou nejčastěji komerční společnosti, které certifikují klientské žádosti a potvrzují identitu žadatele. Získané informace pak připojují k vydanému certifikátu. V dnešní době se používají různé úrovně ověření majitele domény. Od jednoduchého potvrzení odkazu v zaslaném e-mailu (tzv. ověření domény) až po detailnější autorizaci včetně telefonického ověření.
- Nejznámější komerční certifikační autority: Thawte, Symantec (dříve VeriSign), GeoTrust, Comodo, Trustwave

Doplňující informace

- Adresy stránek zabezpečených pomocí SSL začínají https://. Prohlížeč také zabezpečené stránky označuje ikonkou zámku ve stavové liště. Moderní prohlížeče zobrazují ikonku zámku rovněž v řádku adresy a podbarvují tuto řádku různými barvami (zelená pro plně vyhovující, žlutá nebo oranžová pro částečně vyhovující (např. vyhovující certifikát, ale vydaný pro jinou doménu), červená pro nevyhovující certifikát).
- Standardní port pro komunikaci přes HTTPS/SSL je 443, standardní port HTTP je 80.
- HTTPS/SSL dokáže zajistit důvěrnost dat jen na cestě od klienta k serveru (a naopak). Je na provozovateli serveru, jak potom s důvěrnými daty po rozšifrování naloží. Výjimkou není uložení v nešifrované podobě do nechráněné databáze.

Zdroje informací:

www.root.cz/serialy/jak-na-openssl/

cs.wikipedia.org

obrázky vyhledávány na Wikimedia Commons

obrázky vyhledávány na Google - obrázky

A.14 SPDY

SPDY

(Relační vrstva)

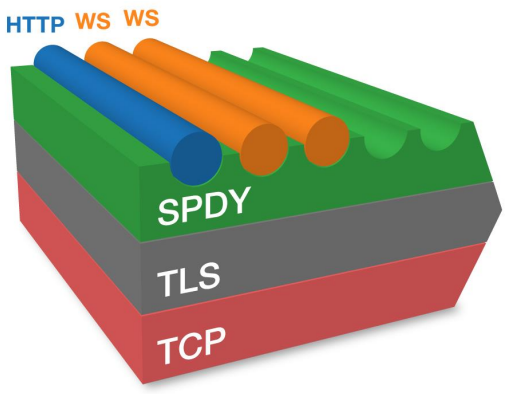
SPDY je experimentální síťový protokol, který je vyvíjen jako součást otevřeného projektu Chromium pod záštitou Googlu. Protokol je součástí návrhu nového standardu HTTP/2.0, který by měl být dokončen na konci roku 2014. Hlavním cílem protokolu je snaha o zajištění rychlejšího načítání webových stránek pomocí úpravy funkčnosti protokolu HTTP. SPDY relace je realizována uvnitř spolehlivého transportního protokolu, například TCP a pro své fungování vyžaduje implementaci na klientské i serverové straně. SPDY je implementován v prohlížečích Chromium/Chrome, Opera, Mozilla Firefox a Internet Explorer. Protokol je implementován, také v jejich mobilních proťějšcích. Autoři naměřili až 64% zkrácení doby potřebné k načtení stránky a při úpravě parametrů TCP až 78%. Průměr zrychlení byl 29%.

Problémy s rychlostí protokolu HTTP

- Přes otevřené spojení se přenese pouze jeden dotaz. HTTP nepřenáší elementy paralelně a otevřený TCP kanál čeká na přijetí odpovědi. Dosavadní řešení problému prohlížeči je otevření několika TCP spojení s daným serverem zároveň.
- Pouze klient rozhodne která data si vyžádá. HTTP server nemá prostředky pro posílání dat klientovi, aniž by si je klient vyžádal. Při komunikaci jsou případy kdy server ví, že klient bude potřebovat určitá data, ale nemůže je doručit, ani klienta informovat o jejich dostupnosti.
- Nekomprimované hlavičky dotazů. Hlavičky mohou mít velikost až 2KB. Moderní webové aplikace používají více cookies a webová odezva pomalých spojení značně trpí.
- Nadbytečné hlavičky. Některé hlavičky jsou v jednom spojení posílány vícekrát, přestože informace v nich obsažené jsou často statické a nepotřebují být obnovovány.
- Nepovinná komprese dat. HTTP volitelně podporuje kompresi dat, ale ne všechny webové servery jí používají.

Vlastnosti

- SPDY přidává relaci nad vrstvou SSL, která umožňuje více multiplexovaných spojení skrze jedno TCP spojení. Syntaxe HTTP metod GET a POST zůstává nezměněna pouze se definuje nový rámec pro přenos dat. SPDY poskytuje několik povinných a nepovinných funkcionalit.



Povinné funkcionality

Multiplexované spojení:

SPDY podporuje neomezený počet souběžných toků dat skrze jediné TCP spojení. Efektivita spojení je maximalizována, protože jsou jednotlivé dotazy posílány zároveň. Sníží se tím také počet TCP spojení k jednomu web serveru.

Priorita dotazování:

S multiplexem je spojen problém priority. Při pomalém spojení může dojít k zadržení paketů, které klient nutně potřebuje. SPDY implementuje prioritu dotazů (úrovně 0 až 7), která tento problém efektivně řeší.

Komprimace hlaviček:

Komprimace hlaviček je vždy zapnutá a snižuje se tím počet odeslaných dat. Hlavičky jsou vždy komprimovány pomocí komprese zlib.

Nepovinné funkcionality

Server push:

Na rozdíl od HTTP, může sám server začít odesílat data. V hlavičce předá klientovi informaci, že začne odesílat data, která si klient ještě nevyžádal. Toto opatření může značně zrychlit načítání stránek, které klient ještě nenavštívil. Pokud má již klient data v paměti pak je odeslání zbytečné, rozhodnutí o odeslání dat náleží jenom serveru, jelikož protokol neposkytuje informace o datech která jsou uloženy u klienta.

Server hint:

Server má možnost, místo aktivního odesílání dat, pouze informovat klienta o potřebných datech. Klient pak může rychleji zareagovat vlastním dotazem. Při pomalém spojení klient rychleji zjistí, která data potřebuje, ještě před tím než by se mu stáhl předchozí dotaz.

Zdroje informací:

http://dev.chromium.org/spdy/spdy-whitepaper
cs.wikipedia.org
obrázky vyhledávány na Wikimedia Commons
obrázky vyhledávány na Google - obrázky

A.15 Transmission Control Protocol

Transmission Control Protocol - TCP

(Transportní vrstva)

- **TCP** protokol (**Transmission Control Protocol**) je jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu. Použitím TCP mohou aplikace na počítačích propojených do sítě vytvořit mezi sebou *spojení*, přes které mohou přenášet data. Protokol garantuje spolehlivé doručování a doručování ve správném pořadí. TCP také rozlišuje data pro vícenásobné, současně běžící aplikace (například webový server a emailový server) běžící na stejném počítači.
- TCP podporuje mnoho na internetu populárních aplikačních protokolů a aplikací, včetně WWW, emailu a SSH.

Technický popis

- TCP je spojově orientovaný protokol pro přenos toku bajtů na transportní vrstvě se spolehlivým doručováním. V současnosti je zdokumentován v IETF RFC 793.
- V sadě protokolů Internetu je TCP prostřední vrstvou mezi IP protokolem pod ním a aplikací nad ním. Aplikace ke vzájemné komunikaci využívají spolehlivé spojení na způsob roury, zatímco IP protokol neposkytuje takové streamy ale jen nespolehlivé pakety. TCP používá služby IP protokolu opakovaným odesíláním nespolehlivých paketů při ztrátě paketu zajišťuje spolehlivost a přeuspořádáváním přijatých paketů zajišťuje správné pořadí. Tím TCP plní úlohu transportní vrstvy ve zjednodušeném modelu ISO/OSI počítačové sítě.

<table>
<tr><td>Bity</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td colspan="16">zdrojový port</td><td colspan="16">cílový port</td></tr>
<tr><td>32</td><td colspan="32">číslo sekvence</td></tr>
<tr><td>64</td><td colspan="32">potvrzený bajt</td></tr>
<tr><td>96</td><td colspan="4">offset dat</td><td colspan="6">rezervováno</td><td colspan="6">příznaky</td><td colspan="16">okénko</td></tr>
<tr><td>128</td><td colspan="16">kontrolní součet</td><td colspan="16">Urgent Pointer</td></tr>
<tr><td>160</td><td colspan="32">volby (volitelné)</td></tr>
<tr><td>192</td><td colspan="24">volby (pokračování)</td><td colspan="8">výplň (do 32)</td></tr>
<tr><td>224</td><td colspan="32">data</td></tr>
</table>

- Aplikace posílá proud (stream) bajtů TCP protokolu k doručení sítí, TCP rozděluje proud bajtů do přiměřeně velkých segmentů. (Velikost segmentů je určena parametrem MTU (*maximum transmission unit*) linkové vrstvy sítě, ke které je počítač připojen.) TCP pak předá takto vzniklé pakety IP protokolu k přepravě internetem do TCP modulu na druhé straně TCP spojení. TCP ověří, že se pakety neztratily tím, že každému paketu přidělil *pořadové číslo*, které se také použije k ověření, že data byla přijata ve správném pořadí.
- TCP modul na straně příjemce posílá zpět *potvrzení* pro pakety které byly úspěšně přijaty. Pokud by se odesilateli potvrzení nevrátilo do rozumné doby (round-trip time, RTT), vypršel by odesilatelův časovač a (pravděpodobně ztracená) data by vyslal znovu.
- TCP protokol ověřuje, zda přenesená data nebyla poškozena šumem tím, že před odesláním spočte kontrolní součet, uloží jej do odesílaného paketu a příjemce kontrolní součet vypočte znovu a ověří, že se shodují.

Jak protokol funguje

TCP porty

K rozlišení komunikujících aplikací používá TCP protokol *čísla portů*. Každá strana TCP spojení má přidruženo 16bitové bezznaménkové číslo portu (existuje 65535 portů) přidělené aplikaci. Porty jsou rozčleněny do třech skupin: dobře známé, registrované a dynamické/privátní. Seznam dobře známých portů je přiřazován organizací Internet Assigned Numbers Authority (IANA) a jsou typicky používané systémovými procesy. Dobře známé aplikace běžící jako servery a pasivně přijímající spojení typicky používají tyto porty. Několik příkladů: FTP (port 21 a 20), SMTP (port 25), DNS (port 53) a HTTP (port 80). Registrované porty jsou typicky používané aplikacemi koncových uživatelů při otevírání spojení k serverům jako libovolná čísla zdrojových portů, ale také mohou identifikovat služby. Dynamické/privátní porty mohou být také používány koncovými aplikacemi, ale není to obvyklé.

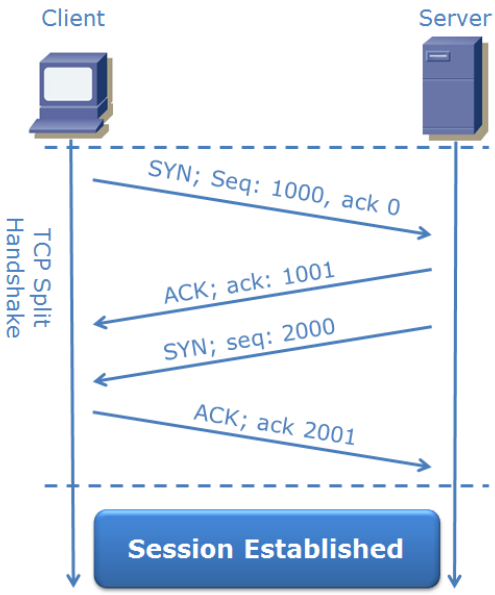
Navázání a ukončení spojení:

Protože TCP je spojovaná transportní služba, musí se před odesíláním dat navázat spojení mezi klientem a serverem. K tomu slouží **trojcestný handshaking** (three-way handshake). V průběhu navazování spojení se obě strany dohodnou na **číslu sekvence** a **potvrzovacím čísle**. Pro navázání spojení se odesílají datagramy s nastavenými příznaky SYN a ACK.

Navázání spojení probíhá ve třech krocích:

- Klient odešle na server datagram s nastaveným příznakem SYN a náhodně vygenerovaným číslem sekvence (x), potvrzovací číslo=0.
- Server odešle klientovi datagram s nastavenými příznaky SYN a ACK, potvrzovací číslo=x+1, číslo sekvence je náhodně vygenerované (y)
- Klient odešle datagram s nastaveným příznakem ACK, číslo sekvence=x+1, číslo odpovědi=y+1.

Obě strany si pamatují číslo sekvence své i protistrany. Používají se totiž i pro další komunikaci a určují pořadí paketů. Když úspěšně proběhne trojcestný handshaking, je spojení navázáno a zůstane tak až do ukončení spojení. To se může zneužít na SYN flood útok.



Ukončení spojení probíhá podobně jako jeho navázání. Používá se k tomu příznaků FIN a ACK:

- Klient odešle datagram s nastaveným příznakem FIN
- Server odpoví datagramem s nastaveným příznakem ACK
- Server odešle datagram s nastaveným příznakem FIN
- Klient odpoví s nastaveným příznakem ACK

Teprve po těchto čtyřech krocích je spojení ukončeno.

Porovnání TCP s jinými protokoly

- Pro mnoho aplikací není TCP vhodné. Velkým problémem je (alespoň u normálních implementací), že aplikace po ztrátě jednoho paketu nemůže dostat následující pakety do té doby, dokud není ztracený paket znovu poslán a úspěšně přijat. To způsobuje problémy realtimeovým aplikacím jako streamovaná média (např. internetové rádio), realtimeové multiplayerové hry a VoIP, kde je často užitečnější dostávat data včas, než je dostávat ve správném pořadí a kompletní.
- Složitost TCP může být problém také pro vestavěná zařízení (embedded systems). Nejlépe známým příkladem je bootování po síti, které obecně používá TFTP (viz PXE). Navíc pro některé triky, jako je přenos dat mezi dvěma uzly, které jsou oba za NATem (použitím STUN nebo podobných protokolů), je mnohem jednodušší, když vám v cestě nestojí složitý protokol jako TCP.
- Tam, kde je TCP nevhodné, se často používá UDP, které poskytuje aplikaci kontrolu/ovládání nad multiplexováním a ověřováním kontrolních součtů. Zato ale UDP neprovádí fragmentaci proudu dat do paketů a zpátky jejich rekonstruování, ani opětovné posílání ztracených paketů. To dovoluje vývojáři aplikace napsat si uvedené funkce tak, jak vyhovuje jeho potřebám nebo je nahradit metodami jako dopředné opravování chyb (*forward error correction*) nebo interpolace.
- SCTP je další IP protokol, který poskytuje spolehlivé, proudově orientované služby nepříliš odlišné od TCP. Je to novější a mnohem složitější protokol než TCP, takže se ještě nedočkal širokého nasazení, ačkoliv je obzvláště navržený k tomu, aby byl používaný v situacích, kdy jsou spolehlivost a téměř real-time ohledy důležité.

Zdroje informací:

www.networksorcery.com/enp/protocol/tcp.htm
cs.wikipedia.org
obrázky vyhledávány na Wikimedia Commons
obrázky vyhledávány na Google - obrázky

A.16 User Datagram Protocol

User Datagram Protocol – UDP

(Transportní vrstva)

- **UDP** (*User Datagram Protocol*) je jeden ze sady protokolů internetu. O protokolu UDP říkáme, že nedává záruky na datagramy, které přenáší mezi počítači v síti. Někdy je označován jako *nespolehlivý*, ale jde o velmi zavádějící označení. Na rozdíl od protokolu TCP totiž nezaručuje, zda se přenášený datagram neztratí, zda se nezmění pořadí doručených datagramů nebo zda nebude některý datagram nedoručen vícekrát.
- Protokol UDP je vhodný pro nasazení, které vyžaduje jednoduchost nebo pro aplikace pracující systémem otázka-odpověď (např. DNS, sdílení souborů v LAN). Jeho bezstavovost je užitečná pro servery, které obsluhují mnoho klientů nebo pro nasazení, kde se počítá se ztrátami datagramů a není vhodné, aby se ztrácel čas novým odesíláním (starých) nedoručených zpráv (např. VoIP, online hry).

Popis protokolu

- **UDP protokol** je protokol transportní vrstvy orientovaný na zprávy a je zdokumentovaný v IETF RFC.
- V sadě protokolů Internetu poskytuje UDP velmi jednoduché rozhraní mezi síťovou vrstvou pod a aplikační vrstvou nad. UDP neposkytuje žádné záruky doručení a odesilatelova UDP vrstva si u jednou už odeslaných zpráv neudržuje žádný stav. UDP pouze přidává kontrolní součty a schopnost roztřiďovat UDP pakety mezi více aplikací běžících na stejném počítači.

+	bity 0 - 15	16 - 31
0	zdrojový port	cílový port
32	délka	kontrolní součet
64	data	

UDP hlavička se skládá jen ze 4 políček, z nichž 2 jsou volitelná. Políčka zdrojového a cílového portu jsou šestnáctibitová a identifikují odesílající a přijímající proces. Protože UDP je bezestavový a odesilatel nemusí vyžadovat odpověď, zdrojový port je volitelný. Pokud není použit, zdrojový port by měl být nastaven na nulu. Po číslech portů následuje povinná délka UDP paketu včetně dat, v bytech. Minimální hodnota činí 8 bajtů. Zbývající políčko hlavičky tvoří šestnáctibitový kontrolní součet pokrývající hlavičku i data. Tento součet je možné vynechat, ale v praxi se téměř vždy používá.
Kvůli chybějícím zárukám se UDP aplikace musí smířit s nějakými ztrátami, chybami nebo duplikacemi. Některé aplikace (jako třeba TFTP) mohou podle potřeby přidávat jednoduchý mechanismus spolehlivosti do aplikační vrstvy. Aplikace používající UDP nejčastěji opravný mechanismus nepotřebují, a dokonce jím mohou být zdržovány. Pokud aplikace vyžaduje vysoký stupeň spolehlivosti, může se místo něj použít TCP nebo opravné kódy.

Porty

- UDP používá porty, aby bylo možné rozlišit v počítači jednotlivé aplikace a správně jim doručit data, i když jich komunikuje v počítači více. Port je 16 bitová hodnota, která umožňuje používat porty z rozsahu 0-65535. Port 0 je rezervován, ale je možné ho použít, pokud odesílající proces neočekává žádnou odpověď.
- Porty 1-1023 jsou tzv. dobře známé (well known ports) a na Unixech a odvozených operačních systémech jsou potřeba práva uživatele root, aby je bylo možné použít. Porty 1024-49151 jsou registrované porty. Porty 49152-65535 jsou používány pro komunikaci klienta se serverem.

Porovnání protokolů TCP a UDP

- TCP je spojově orientovaný protokol což znamená, že k navázání "end-to-end" komunikace potřebuje, aby proběhl mezi klientem a serverem tzv. "handshaking". Poté, co bylo spojení navázáno, data mohou být posílána oběma směry. Charakteristické vlastnosti TCP protokolu jsou:
- spolehlivost – TCP používá potvrzování o přijetí, opětovné posílání a překročení časového limitu. Pokud se jakákoliv data ztratí po cestě, server si je opětovně vyžádá. U TCP nejsou žádná ztracená data, jen pokud několikrát po sobě vyprší časový limit, tak je celé spojení ukončeno.
- zachování pořadí – Pokud pakety dorazí ve špatném pořadí, TCP vrstva příjemce se postará o to, aby se některá data pozdržela a finálně je předala správně seřazená.
- vyšší režie – TCP protokol potřebuje např. tři pakety pro otevření spojení, umožňuje to však zaručit spolehlivost celého spojení.
- UDP je jednodušší protokol založený na odesílání nezávislých zpráv. Charakteristika protokolu:
- bez záruky – Protokol neumožňuje ověřit, jestli data došla zamýšlenému příjemci. Datagram se může po cestě ztratit. UDP nemá žádné potvrzování, přeposílání ani časové limity. V případě potřeby musí uvedené problémy řešit vyšší vrstva.
- nezachovává pořadí – Při odeslání dvou zpráv jednomu příjemci nelze předvídat, v jakém pořadí budou doručeny.
- jednoduchost – Nižší režie než u TCP (není zde řazení, žádné sledování spojení atd.).

Tabulkové porovnání protokolů

TCP, UDP and SCTP Comparison

Attribute	TCP	UDP	SCTP
Reliability	Reliable	Unreliable	Reliable
Connection Management	Connection-orientated	Connectionless	Connection-orientated
Transmission	Byte-orientated	Message-orientated	Message-orientated
Flow Control	Yes	No	Yes
Congestion Control	Yes	No	Yes
Fault Tolerance	No	No	Yes
Data Delivery	Strictly Ordered	Unordered	Partially Ordered
Security	Yes	Yes	Improved

Copyright Technology Training Limited

Zdroje informací:

cs.wikipedia.org
obrázky vyhledávány na Wikimedia Commons
obrázky vyhledávány na Google - obrázky
vlastní zpracování

A.17 Datagram Congestion Control Protocol

Datagram Congestion Control Protocol - DCCP

(Transportní vrstva)

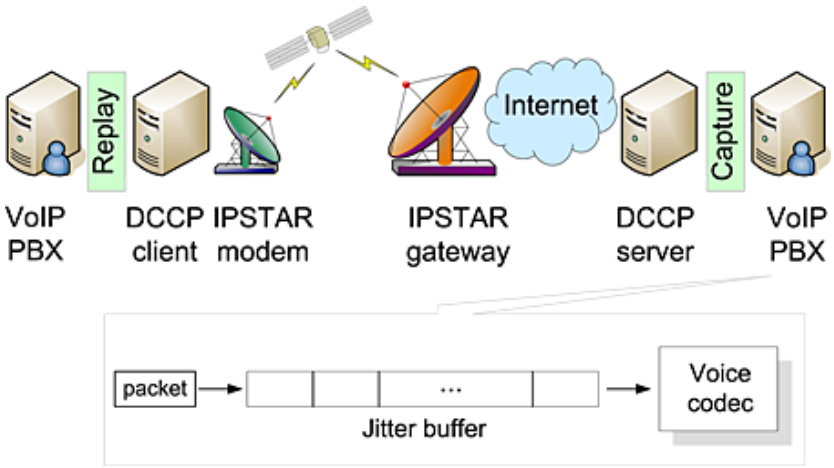
- **DCCP** (**Datagram Congestion Control Protocol**, doslova *datagramový protokol s řízením zahlcení*) je datagramový protokol transportní vrstvy postavený nad IP protokolem.
- Aplikace, které používají protokol DCCP, vyžadují včasné doručení paketů, ale nevyžadují spolehlivé doručení paketů ani správné pořadí přijatých paketů. Jde například o aplikace pracující se streamovanými médii (např. on-line videa, internetová rádia, apod.) nebo internetovou telefonií. Důležitou roli hraje u tohoto protokolu řízení zahlcení, což je mechanismu, který zabraňuje ucpání přenosových cest. Hlavní motivací pro vývoj DCCP je poskytnout přístup ke standardním mechanizmům kontroly zahlcení, bez nutnosti implementovat je v aplikační vrstvě.

- Protokol DCCP je určený pro aplikace, které vyžadují sémantiku TCP, ale nepotřebují doručovací a spolehlivostní mechanizmy TCP, nebo vyžadují jiné dynamické vlastnosti než poskytuje TCP. Podobně je DCCP určen i pro aplikace, které nevyžadují rysy SCTP, například sekvenční doručování vícenásobných streamů (multiple stream).
- Pokud by aplikace pracující se streamovanými médii použila na transportní vrstvě protokol TCP, tak by při ztrátě paketu docházelo k velkým zpožděním, jelikož při ztrátě jednoho paketu se pozastaví zpracování všech dalších paketů dokud nejsou přeposlána data ze ztraceného paketu (TCP garantuje doručení dat ve správném pořadí).

Dodnes měla většina takovýchto aplikací na výběr používat TCP s jeho problémy popsanými výše, nebo používá UDP s vlastní implementací mechanizmu kontroly zahlcení (nebo bez mechanizmu kontroly zahlcení). Příkladem takového protokolu je protokol RTP/RTCP. Účel DCCP je poskytnout standardní cestu k implementaci mechanizmu kontroly zahlcení pro aplikace, které vyžadují kontrolu zahlcení. Jedna z motivací je umožnit použití ECN na obou stranách spojení pro aplikace, které by jinak používaly UDP. Dále DCCP umožňuje spolehlivé sestavení spojení, přátelské ukončení spojení a dohadování o vlastnostech spojení.

- DCCP spojení obsahuje potvrzovaný provoz, stejně jako datový provoz. Potvrzovací pakety informují odesílatele, zda byly jeho datové pakety přijaty, zda byly poškozeny, zda byly zahozeny, zda byly označeny ECN případně jestli aplikace zvládá data přijímat. Potvrzovací pakety jsou odesílány co nejspolehlivěji, tak jak to vyžaduje použitá kontrola zahlcení, včetně možné úplné spolehlivosti doručení potvrzovacích paketů.
- DCCP byl publikován ve standardním doporučení RFC 4340 od IETF v březnu 2006.
- DCCP v sobě neobsahuje žádné bezpečnostní mechanismy, ale jeho přenos lze zabezpečit na nižší vrstvě pomocí IPSec nebo na aplikační vrstvě pomocí DTLS.
- Linux má implementaci DCCP ve svém jádře od verze 2.6.14 a jeho podpora se v každé verzi zlepšuje.

Příklad využití DCCP protokolu



Zdroje informací:

www.read.cs.ucla.edu/dccp/
cs.wikipedia.org
obrázky vyhledávány na Wikimedia Commons
obrázky vyhledávány na Google - obrázky

A.18 Transmission Control Protocol/Internet Protocol

Transmission Control Protocol/Internet Protocol – TCP/IP

Rodina protokolů **TCP/IP** (*Transmission Control Protocol/Internet Protocol* – „primární přenosový protokol/protokol síťové vrstvy“) obsahuje sadu protokolů pro komunikaci v počítačové síti a je hlavním protokolem celosvětové sítě Internet. Komunikační protokol je množina pravidel, které určují syntaxi a význam jednotlivých zpráv při komunikaci.

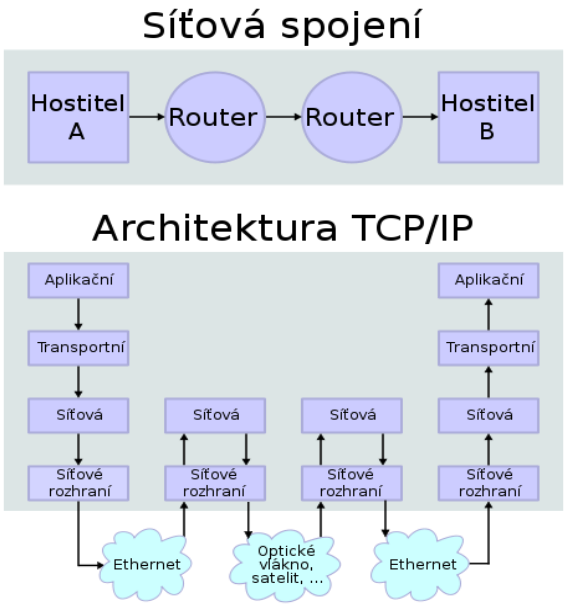
Architektura TCP/IP

Vzhledem ke složitosti problémů je síťová komunikace rozdělena do tzv. vrstev, které znázorňují hierarchii činností. Výměna informací mezi vrstvami je přesně definována. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší.

Komunikace mezi stejnými vrstvami dvou různých systémů je řízena **komunikačním protokolem** za použití spojení vytvořeného sousední nižší vrstvou. Architektura umožňuje výměnu protokolů jedné vrstvy bez dopadu na ostatní. Příkladem může být možnost komunikace po různých fyzických médiích - ethernet, optické vlákno, sériová linka.

Architektura TCP/IP je členěna do čtyř vrstev (na rozdíl od referenčního modelu OSI se sedmi vrstvami):

- aplikační vrstva (*application layer*)
- transportní vrstva (*transport layer*)
- síťová vrstva (*internet layer*)
- vrstva síťového rozhraní (*network interface*)



Vrstva síťového rozhraní:

Nejnižší vrstva umožňuje přístup k fyzickému přenosovému médiu. Je specifická pro každou síť v závislosti na její implementaci. Příklady sítí: Ethernet, Token ring, FDDI, X.25, SMDS.

Síťová vrstva:

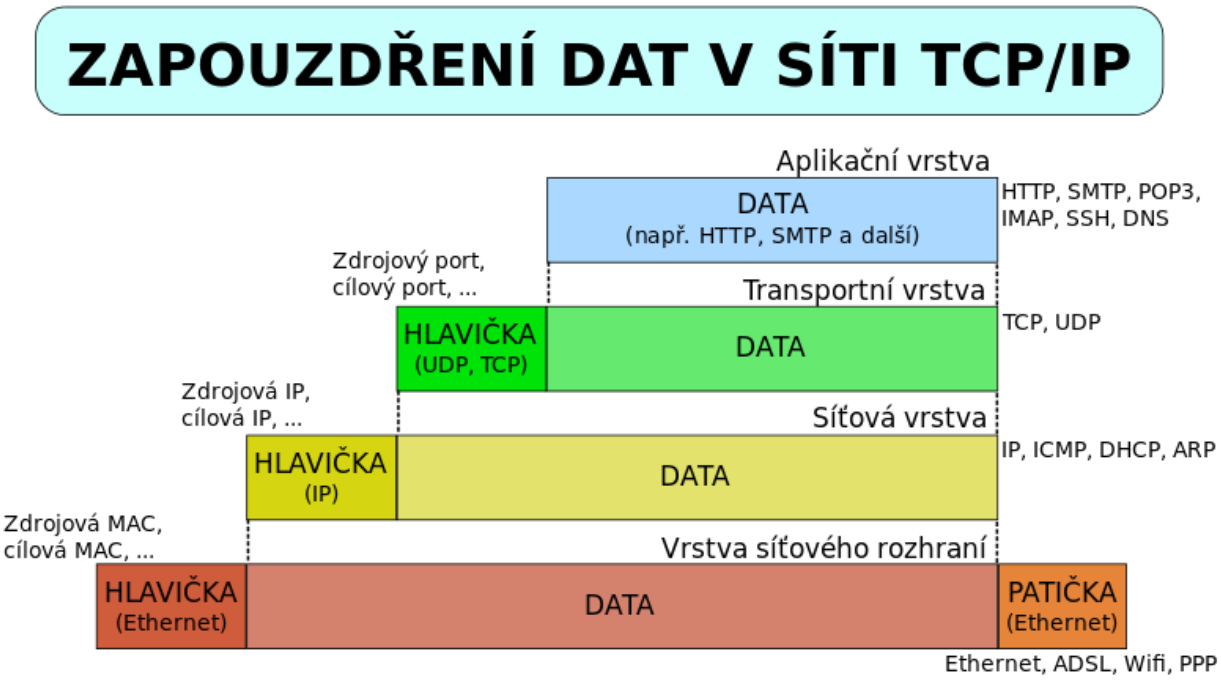
Vrstva zajišťuje především síťovou adresaci, směrování a předávání datagramů. Protokoly: IP, ARP, RARP, ICMP, IGMP, IGRP, IPSEC. Je implementována ve všech prvcích sítě - směrovačích i koncových zařízeních.

Transportní vrstva:

Transportní vrstva je implementována až v koncových zařízeních (počítačích) a umožňuje proto přizpůsobit chování sítě potřebám aplikace. Poskytuje transportní služby kontrolovaným spojením spolehlivým protokolem TCP (*transmission control protocol*) nebo nekontrolovaným spojením nespolehlivým protokolem UDP (*user datagram protocol*).

Aplikační vrstva:

- Vrstva aplikací. To jsou programy (procesy), které využívají přenosu dat po síti ke konkrétním službám pro uživatele. Příklady: Telnet, FTP, HTTP, DHCP, DNS.
- Aplikační protokoly používají vždy jednu ze dvou základních služeb transportní vrstvy: TCP nebo UDP, případně obě dvě (např. DNS). Pro rozlišení aplikačních protokolů se používají tzv. porty, což jsou domluvená číselná označení aplikací. Každé síťové spojení aplikace je jednoznačně určeno číslem portu a transportním protokolem (a samozřejmě adresou počítače).



Zdroje informací:

cs.wikipedia.org

obrázky vyhledávány na Wikimedia Commons

obrázky vyhledávány na Google - obrázky

A.19 Ethernet

Ethernet

- **Ethernet** je název souhrnu technologií pro lokální počítačové sítě (LAN) z větší části standardizovaných jako IEEE 802.3, které používají kabely s kroucenou dvoulinkou, optické kabely (a dříve i koaxiální kabely) pro komunikaci přenosovými rychlostmi od 10 Mbit/s po 10 Gbit/s. Sítě Ethernet realizují fyzickou a linkovou vrstvu referenčního modelu OSI, takže je možné po nich provozovat jeden nebo více protokolů síťové vrstvy, například AppleTalk, DECnet, IPX/SPX a především protokoly IP a IPv6, které se používají pro služby sítě Internet.
- Ještě před rokem 2000 se Ethernet stal dominantní technologií pro drátové nebo kabelové lokální sítě a prakticky synonymem pro lokální síť (LAN). Používá se nejen pro propojování počítačů, ale i pro datová úložiště, zařízení spotřební elektroniky jako jsou televizní přijímače a herní konzole a také jako drátové rozhraní pro přístupové body WiFi a zařízení pro přístup k Internetu. Pokud zařízení deklaruje, že má připojení na LAN, v naprosté většině případů to znamená, že je vybaveno konektorem 8P8C (RJ-45) pro síť Ethernet s rychlostí 100 nebo 1000 Mbit/s.

Formát rámce

- Formát rámce se popisuje pomocí oktetů, což je osmice bitů. Důvodem je přesnost definice, protože některé počítače mohou pracovat s jinou základní délkou bajtu (např. 4 nebo 10 bitů), což by v počítačových sítích způsobovalo nekompatibility. Níže uvedená tabulka popisuje rámec Ethernet II a 802.3, které se liší využitím jednoho pole pro typ nebo pro délku (vysvětlení je pod tabulkou na následujícím snímku).

Ethernetový rámec

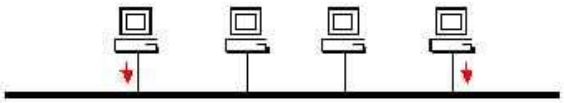
<table>
<tr><td>Preambule</td><td>SFD</td><td>MAC cíle</td><td>MAC zdroje</td><td>Typ/délka</td><td>Data a výplň</td><td>CRC32</td><td>Mezera mezi rámci</td></tr>
<tr><td>7× oktet 10101010</td><td>1× oktet 10101011</td><td>6 oktetů</td><td>6 oktetů</td><td>2 oktety</td><td>46-1500 oktetů</td><td>4 oktety</td><td>12 oktetů</td></tr>
<tr><td colspan="2"></td><td colspan="5">64-1518 oktetů</td><td></td></tr>
<tr><td colspan="7">72-1526 oktetů</td><td></td></tr>
</table>

Popis polí

- Preambule – 7 oktetů, střídavě binární 0 a 1; slouží k synchronizaci hodin příjemce
- SFD – označení začátku rámce (Start of Frame delimiter), oktet 10101011
- MAC cíle – MAC adresa cílového síťového rozhraní o délce 48 bitů; adresa může být individuální (*unicast*), skupinová (*multicast*) a všeobecná (*broadcast*)
- MAC zdroje – MAC adresa zdrojového síťového rozhraní
- Typ/délka
  - pro Ethernet II je to pole určující *typ vyššího protokolu*
  - pro IEEE 802.3 udává délku pole dat; podle IEEE 802.3x má toto pole význam délky pouze pokud obsahuje hodnotu max. 1500 (05DC hex), pokud obsahuje hodnotu alespoň 1536 (0600 hex), jedná se o typ
- Data – pole dlouhé minimálně 46 a maximálně 1500 oktetů (46—1500 B); minimální délka je nutná pro správnou detekci kolizí v rámci segmentu
- Výplň – vyplní zbytek datové části rámce, pokud je přepravovaných dat méně než 46 B
- CRC32 – kontrolní součet (*Frame Check Sequence, FCS*) 32bitový kontrolní kód s generujícím polynomem , který se počítá ze všech polí mezi SFD a FCS; slouží ke kontrole správnosti dat – příjemce si jej vypočítá z obdrženého rámce a pokud výsledek nesouhlasí s hodnotou pole, rámec zahodí jako vadný.

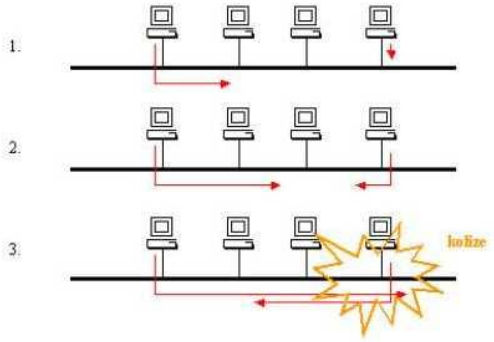
CSMA (Carrier Sense Multiple Access)

- stanice připravená vysílat data si "poslechne" zda přenosové médium (kabel) nepoužívá jiná stanice. V případě, že ano, stanice zkouší přístup později až do té doby dokud není médium volné. V okamžiku kdy se médium uvolní začne stanice vysílat svá data.



CD (Collision Detection)

Stanice během vysílání sleduje zda je na médiu signál odpovídající vysílaným úrovním (tedy aby se např. v okamžiku kdy vysílá signál 0 nevyskytl signál 1). Případ kdy dojde k interakci signálů více stanic se nazývá kolize. V případě detekce kolize stanice generuje signál JAM a obě (všechny) stanice které v daném okamžiku vysílaly generují náhodnou hodnotu času po níž se pokusí vysílání zopakovat.



Kolizní doména

Soubor uzlů jejichž vzájemná činnost může vygenerovat kolizi. Logicky lze odvodit, že kolizní doména by měla být co nejmenší. Používané aktivní prvky mají ke kolizní doméně rozdílný vztah. Některé kolizní doménu rozšiřují, některé kolizní domény oddělují. Jejich volbou lze proto propustnost sítě ovlivnit.

Broadcastová doména

Na počítačové síti se vyskytují principielně dva typy paketů – tzv. unicasty a nonunicasty. Unicasty jsou pakety které mají konkrétního adresáta vyjádřeného regulérní síťovou adresou. Nonunicasty používají skupinovou adresu a jsou určené buď všem uživatelům sítě (broadcasty) nebo vybrané skupině uživatelů (multicasty). Problém je v tom, že nonunicastu se musí počítač věnovat i když není určen pro něj. S nárůstem počtu uzlů v broadcastové doméně narůstá i množství nonunicastů. Z tohoto důvodu je nutné udržet velikost broadcastové domény v rozumné velikosti. Používané aktivní prvky mají k broadcastové doméně rozdílný vztah a proto lze jejich volbou propustnost sítě ovlivnit.

Typy Ethernetu

- **10BASE5** Původní Ethernet na koaxiálním kabelu o rychlosti 10 Mbit/s. Koaxiální kabel o impedanci 50 Ω tvoří sběrnici, ke které se připojují pomocí speciálních transceiverů a AUI kabelů jednotlivé stanice.
- **10BASE2** Ethernet na tenkém koaxiálním kabelu o rychlosti 10 Mbit/s. Koaxiální kabel tvoří sběrnici, ke které se připojují jednotlivé stanice přímo. Kabel je impedance 50 Ω (RG-58) nesmí mít žádné odbočky a je na koncích zakončen odpory 50 Ω (tzv. terminátory).
- **10BASE-T** Jako přenosové médium používá kroucenou dvojlinku s rychlostí 10 Mbit/s. Využívá dva páry strukturované kabeláže ze čtyř. Dnes již překonaná síť, která byla ve většině případů nahrazena rychlejší 100 Mbit/s variantou.
- **10BASE-F** Varianta s optickými vlákny o rychlosti 10 Mbit/s. Používá se pro spojení na větší vzdálenost, nebo spojení mezi objekty, kde nelze použít kroucenou dvojlinku. Tvořila obvykle tzv. **páteřní síť**, která propojuje jednotlivé menší celky sítě. Dnes je již nahrazována vyššími rychlostmi (Fast Ethernet, Gigabit Ethernet).
- **100BASE-TX** Varianta s přenosovou rychlostí 100 Mbit/s, které se říká **Fast Ethernet**, používá dva páry UTP nebo STP kabelu kategorie 5.
- **100BASE-T2** Používá dva páry UTP kategorie 3, 4, 5. Je to varianta vhodná pro starší rozvody strukturované kabeláže.
- **100BASE-T4** Používá čtyři páry UTP kategorie 3, 4, 5. Také vhodná pro starší rozvody strukturované kabeláže.
- **100BASE-FX** Fast Ethernet používající dvě optická vlákna.
- **1000BASE-T** Ethernet s rychlostí 1000 Mbit/s, nazývaný **Gigabit Ethernet**. Využívá 4 páry UTP kabeláže kategorie 5e, je definován do vzdálenosti 100 metrů.
- **1000BASE-CX** Gigabit Ethernet na bázi měděného vodiče pro krátké vzdálenosti, učený pro propojování skupin zařízení.
- **1000BASE-SX** Gigabit Ethernet používající mnohavidové optické vlákno. Je určen pro páteřní sítě do vzdáleností několik set metrů.
- **1000BASE-LX** Gigabit Ethernet používající jednovidové optické vlákno. Je určen pro větší vzdáleností až několika desítek kilometrů.
- **10GBASE-T** Ethernet s rychlostí 10 Gbit/s, nazývaný **Ten Gigabit Ethernet** (nebo také EFM – Ethernet on the first mile). Do vzdálenosti 55 metrů lze využít kabeláž kategorie 6. Pro využití plné délky 100 je nutné použít kategorii 6a (augmented Category 6 – šířka pásma 500 MHz). Někteří výrobci prodávají kabely kategorie 7, které jsou označeny jako kompatibilní s 10GBASE-T.
- **40GBASE** a **100GBASE** s rychlostí 40 a 100 Gbps by měl používat optická vlákna; měděné kabely do délky alespoň 10 metrů

Zdroje informací:

- cs.wikipedia.org
- http://www.svetsiti.cz/
- obrázky vyhledávány pomocí google.cz

A.20 Wi-Fi

Wi - Fi

Bezdrátová komunikace
(*Wireless LAN, WLAN*)

Základní pojmy

- Je to označení pro několik standardů ***IEEE 802.11*** popisujících bezdrátovou komunikaci v počítačových sítích (též *Wireless LAN*, *WLAN*).
- Samotný název ***WiFi*** vytvořilo **Wireless Ethernet Compatibility Aliance**. Tato technologie využívá bezlicenčního frekvenčního pásma, proto je ideální pro budování levné, ale výkonné sítě bez nutnosti pokládky kabelů.
- Název původně neměl znamenat nic, ale časem se z něj stala slovní hříčka *wireless fidelity* (bezdrátová věrnost) analogicky k ***Hi-Fi*** (*high fidelity* – vysoká věrnost).

IEEE 802.11

- Je Wi-Fi standard s dalšími doplňky pro lokální bezdrátové sítě, vyvíjený 11. pracovní skupinou ***IEEE LAN/MAN*** standardizační komise *IEEE 802* (*z angl.* ***Institute of Electrical and Electronic Engineers***). Výraz *802.11x* je používán pro množinu doplňků k tomuto standardu.
- Standard 802.11 zahrnuje několik druhů modulací pro posílání radiového signálu, přičemž všechny používají stejný protokol. Nejpoužívanější modulace jsou definované v dodatcích k původnímu standardu s písmeny *a*, *b*, *g*.
- Standardy 802.11b a 802.11g používají 2,4 gigahertz (GHz) pásmo. Proto mohou zařízení interferovat s mikrovlnnými troubami, bezdrátovými telefony, s Bluetooth nebo s dalšími zařízeními používajícími stejné pásmo. Oproti tomu standard 802.11a používá 5 GHz pásmo a není tedy ovlivněn zařízeními pracujícími v pásmu 2,4 GHz.
- Standard 802.11ac využívá zároveň obě pásma, tedy 2,4 gigahertz (GHz) a zároveň 5 gigahertz (GHz).

Přehled standardů IEEE 802.11

Standard	Rok vydání	Pásmo [GHz]	Maximální rychlost [Mbit/s]	Fyzická vrstva
IEEE 802.11	1997	2,4	2	DSSS a FHSS
IEEE 802.11a	1999	5	54	OFDM
IEEE 802.11b	1999	2,4	11	DSSS
IEEE 802.11g	2003	2,4	54	OFDM
IEEE 802.11n	2009	2.4 nebo 5	600	MIMO a OFDM
IEEE 802.11y	2008	3,7	54	
IEEE 802.11ac	2013	5	1000	MU – MIMO
IEEE 802.11ad	2014	2,4 , 5 , a 60	7000	

Charakteristika

- Původním cílem Wi-Fi sítí bylo zajišťovat vzájemné bezdrátové propojení přenosných zařízení a dále jejich připojování na lokální (např. firemní) sítě LAN. S postupem času začala být využívána i k bezdrátovému připojení do sítě Internet v rámci rozsáhlejších lokalit a tzv. hotspotů. Wi-Fi zařízení jsou dnes prakticky ve všech přenosných počítačích a i v některých mobilních telefonech.

- Úspěch Wi-Fi přineslo využívání bezlicenčního pásma, což má negativní důsledky ve formě silného zarušení příslušného frekvenčního spektra a dále častých bezpečnostních incidentů. Následníkem Wi-Fi by měla být bezdrátová technologie WiMAX, která se zaměřuje na zlepšení přenosu signálu na větší vzdálenosti.

- Wi-Fi zajišťuje komunikaci na spojové vrstvě, zbytek je záležitost vyšších protokolů (na rozdíl od Bluetooth, který sám o sobě zajišťuje nejrůznější služby). Typicky se proto přenášejí zapouzdřené ethernetové rámce. Pro bezdrátovou komunikaci na sdíleném médiu (*šíření elektromagnetického pole prostorem*) je používán protokol CSMA/CA (*Ethernet používá na vodičích CSMA/CD*).

Komponenty Wi-Fi

- **Access point (AP)** je prvek, který umožňuje vysílat nebo přijímat data. AP jsou stěžejními prvky pro sítě WLAN. Hlavní AP vysílají pomocí všesměrových nebo směrových antén signál do širokého okolí a tento signál je přijímán AP jednotlivých uživatelů.
- **Anténa -** v mnoha případech je potřeba pro kvalitní příjem nutná **anténa**. Rozeznáváme všesměrové a směrové antény. Všesměrové antény jsou vhodné pro pokrytí velké oblasti WiFi signálem, zatímco pomocí směrových antén můžete přenášet WiFi signál na velké vzdálenosti.
- **WiFi router** v sobě kombinuje funkci klasického routeru a AP. Většinou je takový router vybaven jedním portem WAN (Wide Area Network), několika ethernetovými porty a anténou nebo anténami pro bezdrátovou komunikaci. Pomocí WiFi routeru si můžete snadno vytvořit svou domácí bezdrátovou síť.
- **WiFi karta**, ať už do PCI nebo do notebookového portu PCMCIA, slouží pro připojení počítače nebo notebooku k WiFi síti podobně jako síťová karta slouží k připojení na LAN. Mnoho notebooků, ale i přenosných zařízení jako jsou různé mobilní telefony a PDA mají zabudován WiFi modul.

Zabezpečení sítě

- Problém bezpečnosti bezdrátových sítí vyplývá zejména z toho, že jejich signál se šíří i mimo zabezpečený prostor bez ohledu na zdi budov, což si mnoho uživatelů neuvědomuje. Dalším problémem je fakt, že bezdrátová zařízení se prodávají s nastavením bez jakéhokoliv zabezpečení, aby po zakoupení fungovala ihned po *zapojení do zásuvky*.
- Nezvaný host se může snadno připojit i do velmi vzdálené bezdrátové sítě jen s pomocí směrové antény, i když druhá strana výkonnou anténu nemá. Navíc většina nejčastěji používaných zabezpečení bezdrátových sítí má jen omezenou účinnost a dá se snadno obejít.
- Různé typy zabezpečení se vyvíjely postupně a proto starší zařízení poskytují jen omezené nebo žádné možnosti zabezpečení bezdrátové sítě. Právě kvůli starším zařízením jsou bezdrátové sítě někdy zabezpečeny jen málo. V takových případech je vhodné použít zabezpečení na vyšší síťové vrstvě, například virtuální privátní síť.
- Útočník vybavený směrovou anténou se může připojit do sítě vzdálené několik set metrů nebo dokonce několik kilometrů, i když síť sama o sobě má dosah pár desítek metrů. Různé typy zabezpečení se vyvíjely postupně, proto starší poskytují pouze omezené možnosti zabezpečení sítě, které se dají snadno prolomit. Neznalí uživatelé mohou také doplatit na to, že bezdrátové zařízení se prodávají bez nastaveného zabezpečení, nebo s nějakým výchozím nastavením, které je u všech zařízení daného typu stejné. Například heslo: „password". Na internetu jsou databáze s tímto nastavením, proto takto zabezpečenou síť dokáže prolomit i běžný uživatel se schopností hledání na googlu.

Rozdělení zabezpečení

Bezpečnost bezdrátových sítí můžeme rozdělit do dvou hlavních skupin:

- **šifrování** = zabezpečení přenášených dat před odposlechem
- **autorizace** = řízení přístupu oprávněných uživatelů

Způsoby zabezpečení

- **Zablokování vysílání SSID -** Zablokování vysílání SSID sice porušuje standard, ale je nejjednodušším zabezpečením bezdrátové sítě pomocí jejího zdánlivého skrytí. Klienti síť nezobrazí v seznamu dostupných bezdrátových sítí, protože nepřijímají broadcasty se SSID. Ovšem při připojování klienta k přípojnému bodu je SSID přenášen v otevřené podobě a lze ho tak snadno zachytit. Při zachytávání SSID při asociaci klienta s přípojným bodem se používá i provokací, kdy útočník do bezdrátové sítě vysílá rámce, které přinutí klienty, aby se znovu asociovali.
- **Kontrola MAC adres -** Přípojný bod bezdrátové sítě má k dispozici seznam MAC adres klientů, kterým je dovoleno se připojit (tzv. whitelist). Zrovna tak je možné nastavit blokování určitých MAC adres (*blacklist*). Útočník se může vydávat za stanici, která je již do bezdrátové sítě připojena pomocí nastavení stejné MAC adresy (pokud je na AP tato funkce aktivní).

- **WEP -** Šifrování komunikace pomocí statických ***WEP*** klíčů (*Wired Equivalent Privacy*) symetrické šifry, které jsou ručně nastaveny na obou stranách bezdrátového spojení. Díky nedostatkům v protokolu lze zachycením specifických rámců a jejich analýzou klíč relativně snadno získat. Pro získání klíčů existují specializované programy.
- **WPA -** Kvůli zpětné kompatibilitě využívá WPA (Wi-Fi Protected Access) WEP klíče, které jsou ale dynamicky bezpečným způsobem měněny. K tomu slouží speciální doprovodný program, který nazýváme *prosebník* (suplikant). Z tohoto důvodu je možné i starší zařízení WPA vybavit. Autentizace přístupu do WPA sítě je prováděno pomocí **PSK** *(Pre-Shared Key – obě strany používají stejnou dostatečně dlouhou heslovou frázi*) nebo RADIUS server (ověřování přihlašovacím jménem a heslem).
- **WPA2 -** Novější WPA2 přináší kvalitnější šifrování (šifra AES), která však vyžaduje větší výpočetní výkon a proto nelze WPA2 používat na starších zařízeních.

Zdroje informací:

- cs.wikipedia.org
- obrázky vyhledávány pomocí google.cz