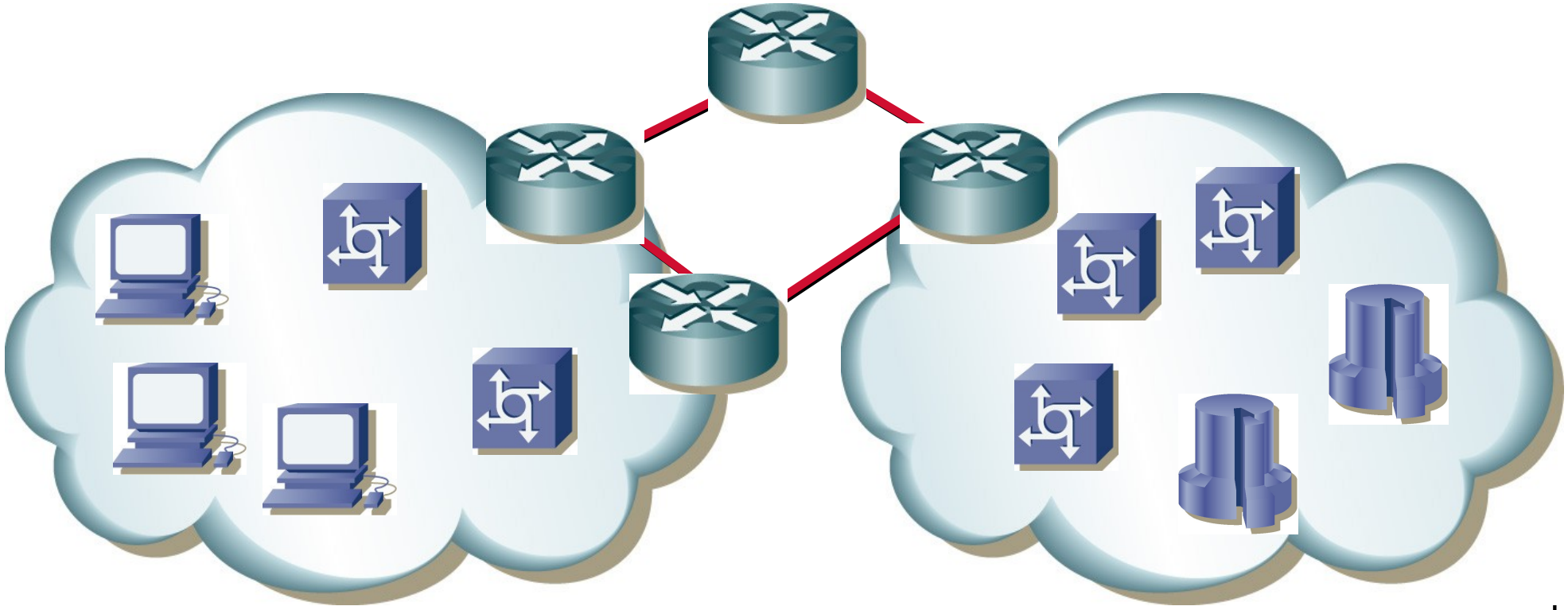
Počítačové sítě II

19. FTP, vzdálené přihlašování, SNMP

Miroslav Spousta, 2006
<qiq@ucw.cz>, http://www.ucw.cz/~qiq/vsfs/



Další protokoly

- kromě „základních“ a nejrozšířenějších protokolů (DNS, SMTP, POP3, IMAP, HTTP) se v Internetu používá řada dalších pro nejrůznější služby
  - FTP, TFTP: přenos souborů
  - SNMP: správa síťových prvků
  - NTP: synchronizace času
  - TELNET, SSH: vzdálené přihlašování
  - NFS, CIFS: sdílení souborů přes síť
  - SIP: internetová telefonie, RTP: přenos audia/videa
  - NNTP: news skupiny
  - ...
- alespoň na některé se podíváme

FTP

- File Transfer Protocol, RFC 959
- protokol pro přenos souborů, používá architekturu klient-server
- textový protokol (podobně jako SMTP, POP3, IMAP, ...)
- využívá spojení TCP, porty 20 a 21
  - FTP server naslouchá na portu 21
- široká podpora mezi klienty
  - file managery a dokonce web browsery mohou fungovat jako FTP klient
  - používá URL ve formátu: ftp://*login*:*heslo*@*serever*:*port*
- poměrně jednoduchý na implementaci
- často se používá pro distribuci souborů, přihlášení pak bývá pomocí anonymního účtu
  - login zpravidla „anonymous“ a heslo e-mailová adresa

Přenos souborů

- datová spojení (přenos souboru, výpis adresáře)
  - při použití aktivního módu server otevírá spojení ke klientovi z portu 20
    - cílová adresa a port se zadává pomocí příkazu PORT
  - při použití pasivního spojení klient otevírá spojení na server na určitý port
    - adresa a port na straně serveru se získá z odpovědi na příkaz PASV
  - porty se zadávají jako dvojice čísel 0-255 (po bajtech)
  - je možné navazovat spojení na jiného klienta/jiný server, než je navázané datové spojení (resp. přenášet data mezi dvěma servery)
- existují dva módy datového spojení: ASCII a BINARY
  - ASCII: při přenosu probíhá konverze konce řádků, BINARY: bez konverze
- chybové kódy jsou obdobné jako u SMTP a http
- příkazy jsou textové, argumenty jsou odděleny mezerou (čárkami)
  - USER (login), PASS (heslo), CWD (aktuální adresář), LIST (výpis aktuálního adresáře), RETR (získání souboru), STOR (nahrání souboru), QUIT (konec), SYST (systém (pro výpis adresáře)), ...

Aktivní a pasivní mód

- Aktivní mód přenosu dat:

- Pasivní mód přenosu dat:

Ukázka FTP spojení

```
S: 220 (vsFTPd 2.0.3)
C: USER qiq
S: 331 Please specify the password.
C: PASS password
S: 230 Login successful.
C: SYST
S: 215 UNIX Type: L8
C: PORT 195,113,21,122,133,132
S: 200 PORT command successful. Consider using PASV.
C: LIST work
<navázání spojení z portu 20 na port 34180>
S: 150 Here comes the directory listing.
<přenos a ukončení spojení na port 34180>
S: 226 Directory send OK.
C: QUIT
S: 221 Goodbye.
```

```
C: PASV
S: 227 Entering Passive Mode (195.113.21.122,70,63)
```

Zabezpečená verze FTP

- v původní verzi FTP se posílá heslo po navázání spojení v otevřené podobě (plain text)
- existují rozšířené verze, které používají šifrovaný kanál
- FTPS:
  - rozšíření FTP o možnost šifrování spojení pomocí SSL/TLS
  - šifrovat se může řídící spojení, případně i datové
  - pro řídící spojení se spouští pomocí AUTH TLS, pro datové pomocí příkazu PROT
  - šifrování pro řídící spojení je možné i zrušit (např. po autentizaci)
- SFTP: secure shell FTP
  - zcela odlišný protokol, pro šifrování se používá SSH a kanály uvnitř spojení (SSH 2)

Ukázka šifrování FTP spojení

```
S: 220 (vsFTPd 2.0.3)
C: FEAT
S: 211-Features:
S:  AUTH TLS
S: 211 End
C: AUTH TLS
<sestavení TLS spojení>
S: 234 Proceed with negotiation.
C: USER test
S: 331 Please specify the password.
C: PASS XXXX
S: 230 Login successful.
C: PBSZ 0
S: 200 PBSZ set to 0.
C: PROT P
S: 200 PROT now Private.
C: PASV
S: 227 Entering Passive Mode (195,113,21,112,57,140)
<navazování spojení na port 14732>
C: LIST
S: 150 Here comes the directory listing.
<sestavení TLS spojení>
<přenos výpisu adresáře po datovém spojení>
-rw-------    1 1001    101         39074 Feb 12 17:18 10.jpg
<uzavření datového spojení>
S: 226 Directory send OK.
C: QUIT
<uzavření řídícího spojení>
```

TFTP

- Trivial File Transfer Protocol, RFC 1350
- velmi jednoduchý protokol pro přenos souborů
  - určený také pro jednoduchá (a „hloupá“) zařízení (X terminál, AP, routery)
  - pro nahrání souborů pro bootování, firmwaru, ...
- používá UDP, port 69
- maximální velikost souboru je 32 MB
- řízení je jednoduché: vždy po síti cestují buďto žádost, data, nebo potvrzení (pomalé pro linky s velkou latencí)
- nepoužívá žádnou autentizaci
- neumožňuje vypisovat obsah adresáře, jen přijmout a odeslat soubor
  - samotné datagramy jsou velmi jednoduché: jen operace, číslo bloku (a data)
  - data se přenáší po blocích velikosti 512 B, konec souboru je indikován datagramem s velikostí (0 – 511 B)
- umožňuje přenášet binární a textová data (podobně jako FTP)

TFTP: ukázka čtení souboru

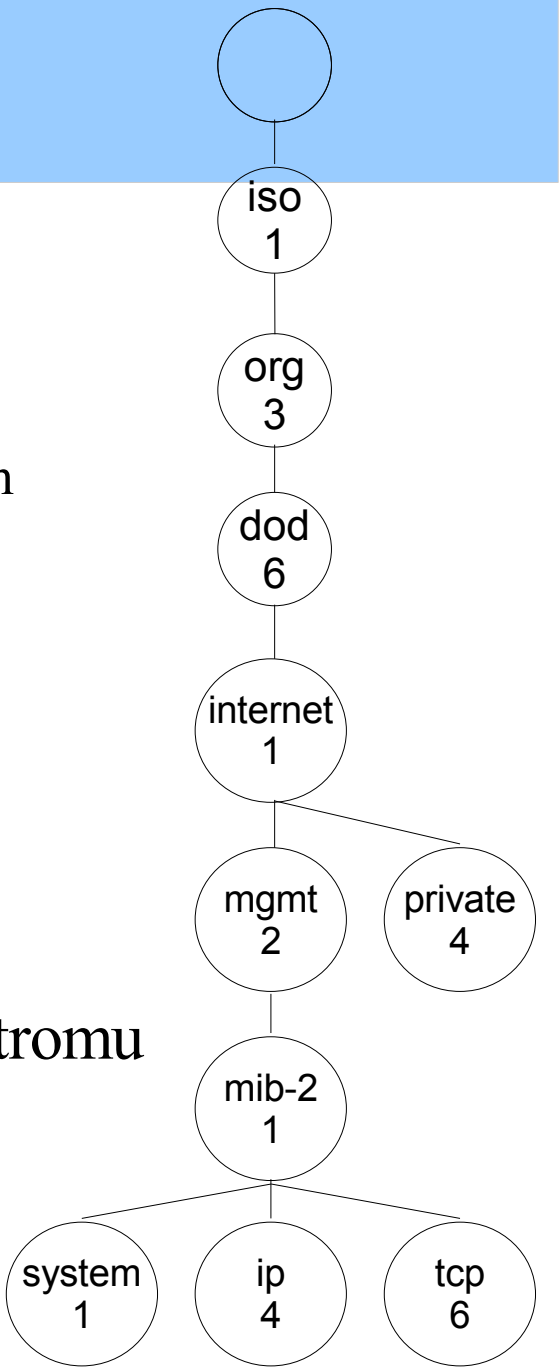
```
sent RRQ <file=test, mode=octet>
received DATA <block=1, 512 bytes>
sent ACK <block=1>
received DATA <block=2, 512 bytes>
sent ACK <block=2>
received DATA <block=3, 512 bytes>
sent ACK <block=3>
received DATA <block=4, 512 bytes>
sent ACK <block=4>
received DATA <block=5, 512 bytes>
sent ACK <block=5>
received DATA <block=6, 512 bytes>
sent ACK <block=6>
received DATA <block=7, 512 bytes>
sent ACK <block=7>
received DATA <block=8, 512 bytes>
sent ACK <block=8>
received DATA <block=9, 512 bytes>
sent ACK <block=9>
received DATA <block=10, 512 bytes>
sent ACK <block=10>
received DATA <block=11, 448 bytes>
```

SNMP

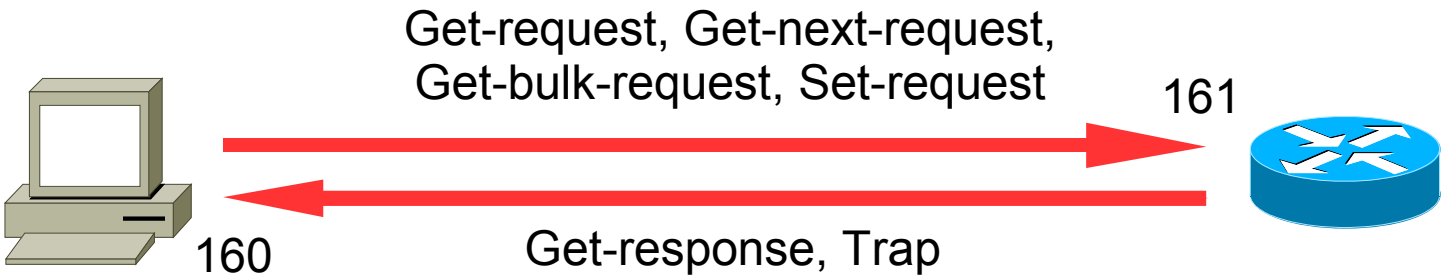
- Simple Network Management Protocol, RFC 3418
- slouží k monitorování a nastavení síťových zařízení
  - routery, switche, tiskárny, …
  - umožňuje získat informace o stavu zařízení, nastavit je
  - zařízení mohou generovat asynchronní události (trap), které jsou zasílány SNMP manageru, který na daný stav může reagovat
- používá UDP, porty 161 a 162
  - zařízení naslouchá na portu 161
  - SNMP manager naslouchá na portu 162
- informace jsou uloženy ve strukturované databázi se stromovou strukturou
  - MIB (Management Information Base)

MIB

- Management Information Base
  - databáze objektů
  - položky jsou identifikovány čísly, dohromady tvoří strom
  - místo čísel se používá také symbolické označení
- struktura podstromu private definována různě
- struktura podstromu mib-2 definována v RFC:
  - TCP: RFC 4022, UDP: RFC 4113, IP: RFC 2011
  - IF (interface): RFC 2863, Fibre Channlel: RFC 4044
- pole hodnot jsou definována jako další úroveň ve stromu
  - hodnoty získáme pomocí Get-next-request



SNMP



- Get-request: získání hodnoty, Get-next-request: získání následující hodnoty (klient nemusí znát všechny MIB hodnoty), Get-bulk-request: získání několika hodnot najednou
- Get-responese: odpověď
- Set-request: nastavení hodnoty
- Trap: asynchronní informace od síťového zařízení

SNMP: příklad

```
qiq@bug:~$ snmpwalk -v1 -c public 195.113.21.100 system
SNMPv2-MIB::sysDescr.0 = STRING: HP ETHERNET MULTI-ENVIRONMENT,ROM
A.05.03,JETDIRECT,JD24,EEPROM A.05.05
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.11.2.3.9.1
SNMPv2-MIB::sysUpTime.0 = Timeticks: (82848620) 9 days, 14:08:06.20
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 64

qiq@bug:~$ snmpwalk -On -v1 -c public 195.113.21.100 system
.1.3.6.1.2.1.1.1.0 = STRING: HP ETHERNET MULTI-ENVIRONMENT,ROM
A.05.03,JETDIRECT,JD24,EEPROM A.05.05
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.11.2.3.9.1
.1.3.6.1.2.1.1.3.0 = Timeticks: (82853602) 9 days, 14:08:56.02
.1.3.6.1.2.1.1.4.0 = STRING:
.1.3.6.1.2.1.1.5.0 = STRING:
.1.3.6.1.2.1.1.6.0 = STRING:
.1.3.6.1.2.1.1.7.0 = INTEGER: 64
```

NTP

- Network Time Protocol, RFC 1305
- používá UDP, port 123
- jeden z nejstarších stále používaných protokolů
- mnoho protokolů používá datum a čas, je žádoucí, aby se tyto údaje na jednotlivých počítačích příliš nelišil
- používá hierarchickou strukturu time serverů
  - na nejvyšší úrovni jsou servery synchronizované s přesným zdrojem času (např. pomocí rádiového vysílání)
  - tzv. stratum 1 servery
  - z nich odvozují čas servery stratum 2, atd.
  - některé time servery jsou veřejné, jiné určené pouze pro lokální síť
  - je možné synchronizovat čas s několika servery
- veřejné NTP servery: http://ntp.isc.org/ (různé povolené použití)
- NTP server pools: http://pool.ntp.org

Telnet

- telnet je služba, která umožňuje obousměrnou komunikaci
  - používaná pro vzdálený textový přístup k různým službám (7bitový, US-ASCII)
  - telnet protokol používá TCP, port 23
  - RFC 854, RFC 855 definují základní verzi, existuje mnoho rozšíření
    - např. echo, 8bitový přenos dat
- telnet je také program, který emuluje terminál
  - je možné ho použít ke TCP komunikaci libovolným textovým protokolem
  - např. POP3, SMTP, …
- problémy:
  - běžně se nepoužívá šifrování komunikace, není možné ověřit protistranu (řešení: SSH)

Telnet: protokol

- kromě US-ASCII znaků je možné přenášet i speciální příkazy
- jsou uvozeny znakem IAC (Interpret As Command) – 0xFF
- BRK (break – stisknuta klávesa pro přerušení), IP (interrupt – přerušení), EC (erase character), EL (erase line)
- WILL, WONT, DO, DONT: pro domlouvání o možnostech (option) přenosu: echo, terminal type, window size, terminal speed, …
- možnosti požadavků a odpovědí:
  - WILL -> DO/DONT (odesilatel chce používat option, protistrana potvrzuje/zakazuje)
  - DO -> WILL/WONT (odesilatel žádá protistranu, aby používala option, protistrana potvrzuje/odmítá)
  - WONT -> DONT (odesilatel nebude používat option, protistrana musí respektovat)
  - DONT -> WONT (odesilatel žádá protistranu, aby nepoužívala option)

Telnet: ukázka výměny options

```
SENT WILL NEW-ENVIRON
SENT WILL AUTHENTICATION
SENT DO SUPPRESS GO AHEAD
SENT WILL TERMINAL TYPE
SENT WILL NAWS
SENT WILL TSPEED
SENT WILL LFLOW
SENT WILL LINEMODE
SENT DO STATUS
SENT WILL XDISPLOC
RCVD DO NEW-ENVIRON
RCVD DO AUTHENTICATION
RCVD IAC SB AUTHENTICATION SEND 7
CLIENT|ONE-WAY
SENT IAC SB AUTHENTICATION IS NULL
CLIENT|ONE-WAY
RCVD WILL SUPPRESS GO AHEAD
RCVD DO TERMINAL TYPE
RCVD DO NAWS
SENT IAC SB NAWS 0 98 (98) 0 31 (31)
RCVD DO TSPEED
RCVD DO LFLOW
```

```
RCVD DONT LINEMODE
RCVD WILL STATUS
RCVD DO XDISPLOC
RCVD IAC SB ENVIRON SEND
SENT IAC SB ENVIRON IS VAR "DISPLAY"
VALUE "localhost.localdomain:0.0"
RCVD IAC SB TERMINAL-SPEED SEND
RCVD IAC SB X-DISPLAY-LOCATION SEND
RCVD IAC SB ENVIRON SEND
SENT IAC SB ENVIRON IS VAR "DISPLAY"
VALUE "localhost.localdomain:0.0"
RCVD IAC SB TERMINAL-TYPE SEND
RCVD DO ECHO
SENT WONT ECHO
RCVD WILL ECHO
SENT DO ECHO
```

Secure SHell

- vznikl jako bezpečná obdoba protokolu telnet
- dnes umožňuje širší použití (přenos souborů, přesměrování portů, přesměrování X session)
  - spíše než shell přístup se dnes jedná o sestavení šifrovaného kanálu mezi dvěma uzly v Internetu
- vznikl v roce 1995, používá TCP, port 22
  - původní verze: SSH 1, nyní se používá SSH 2
  - RFC 4251, zatím není Internet standard
- používá public-key cryptography, ale ne certifikáty
- otisky klíčů serverů se ukládají v souboru
  - při příštích připojeních se kontrolují, změna je hlášena
  - brání se tak man-in-the-middle útokům
- umožňuje různé způsoby ověřování

SSH 2

- SSH 2 se skládá z několika vrstev:
- **transportní**: zajišťuje prvotní výměnu klíčů, nastavuje používanou šifru, autentizaci, poskytuje rozhraní pro posílání datových paketů
- **autentizační**: zajišťuje ověření uživatele pomocí několika různých metod
  - password, public-key, keyboard-interactive, GSSAPI (kerberos)
- **spojení**: vytváří kanály pro různé účely (shell, port forwarding, přenos souboru, …)
  - v jednom spojení může současně běžet mnoho kanálů
- součástí SSH 2 je i protokol pro přenos souborů: SFTP

SSH 2: Port forwarding

- SSH 2 je možné použít pro přesměrování portu z lokálního počítače skrz šifrovaný kanál na jiný počítač
  - poor man's VPN

```
bug$ ssh -L 8080:localhost:80 qiq@host
Password:

You have mail.
host$
```

```
bug$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Date: Sat, 06 May 2006 11:49:37 GMT
...
```

Síťové filesystémy

- často je užitečné sdílet data přes síť
- ideální stav: uživatel neví, zda jsou data uložena lokálně (na disku), nebo přes síťové připojení
- existuje mnoho různých protokolů pro síťový přístup
  - NFS, CIFS, FTP, SFTP, …
  - některé jsou přímo navržené jako síťové filesystémy, jiné slouží hlavně pro přenos dat
- nejrozšířenější: NFS, CIFS

NFS

- Network File systém, RFC 3530 (verze 4)
- jeden z nejstarších síťových filesystémů, klient-server architektura
- původně ze světa UNIXu (Sun Microsystems)
- používá UDP (od verze 3 i TCP)
- původně bezestavový protokol
  - síťová připojení přežijí i např. restart serveru
  - problém se zamykáním
- původně hesla v nešifrované podobě
  - dnes: kerberos autentizace
- často se síťové disky připojovaly automaticky (automount) při přístupu k danému adresáři

CIFS

- SMB (Server Message Block)
  - původně od IBM, pro DOS
  - později Microsoft přejmenoval a rozšířil na CIFS
  - používá porty UDP: 137, 138, TCP: 139, 445 (CIFS)
- opět klient-server architektura, původně určené pro lokální sítě
  - sdílet je možné soubory, ale i tiskárny
- protokol nebyl přijatý jako RFC (problémy s patenty MS)
- pro překlad jména se používá DNS nebo NetBIOS (záleží na konfiguraci)
- různá rozšíření, např.: Unix extensions for CIFS