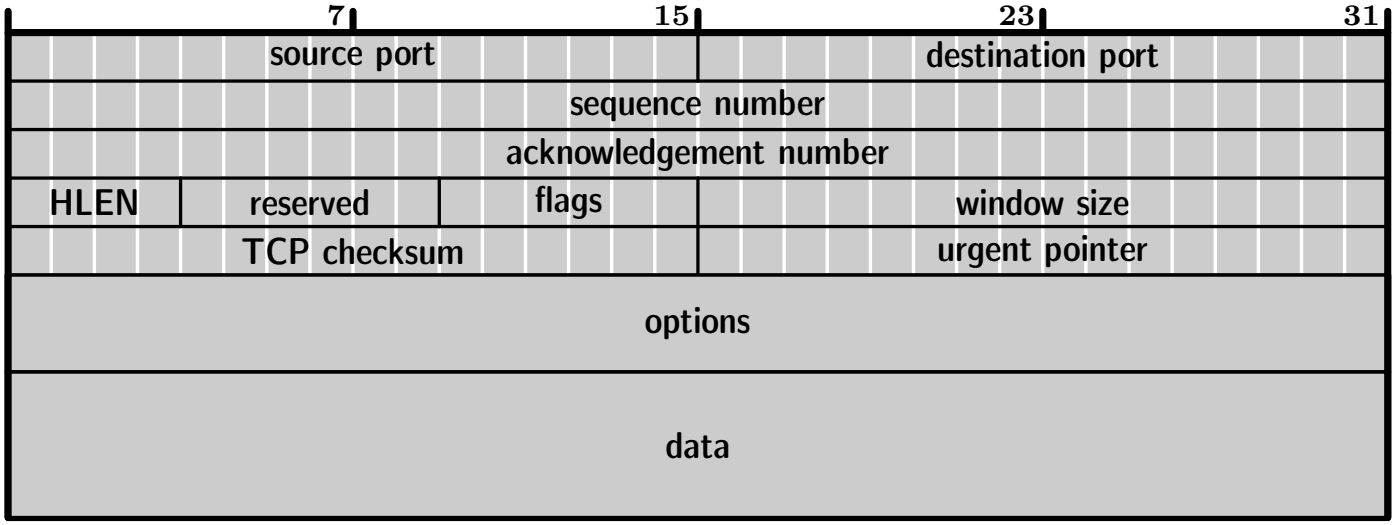
Kvantitativní analýza internetového provozu (2)

Ladislav Lhotka
⟨*lhotka@cesnet.cz*⟩

Osnova přednášky

- TCP – navázání a ukončení spojení
- TCP – algoritmy proti zahlcení
- Útoky proti TCP
- User Datagram Protocol (UDP)
- Příklady typických aplikací: HTTP, DNS, XMPP
- Typický síťový „útok“: port scan
- Časové charakteristiky provozu
- Metody sběru dat

Hlavička TCP



Příznaky (flags): URG ACK PSH RST SYN FIN

Navázání spojení

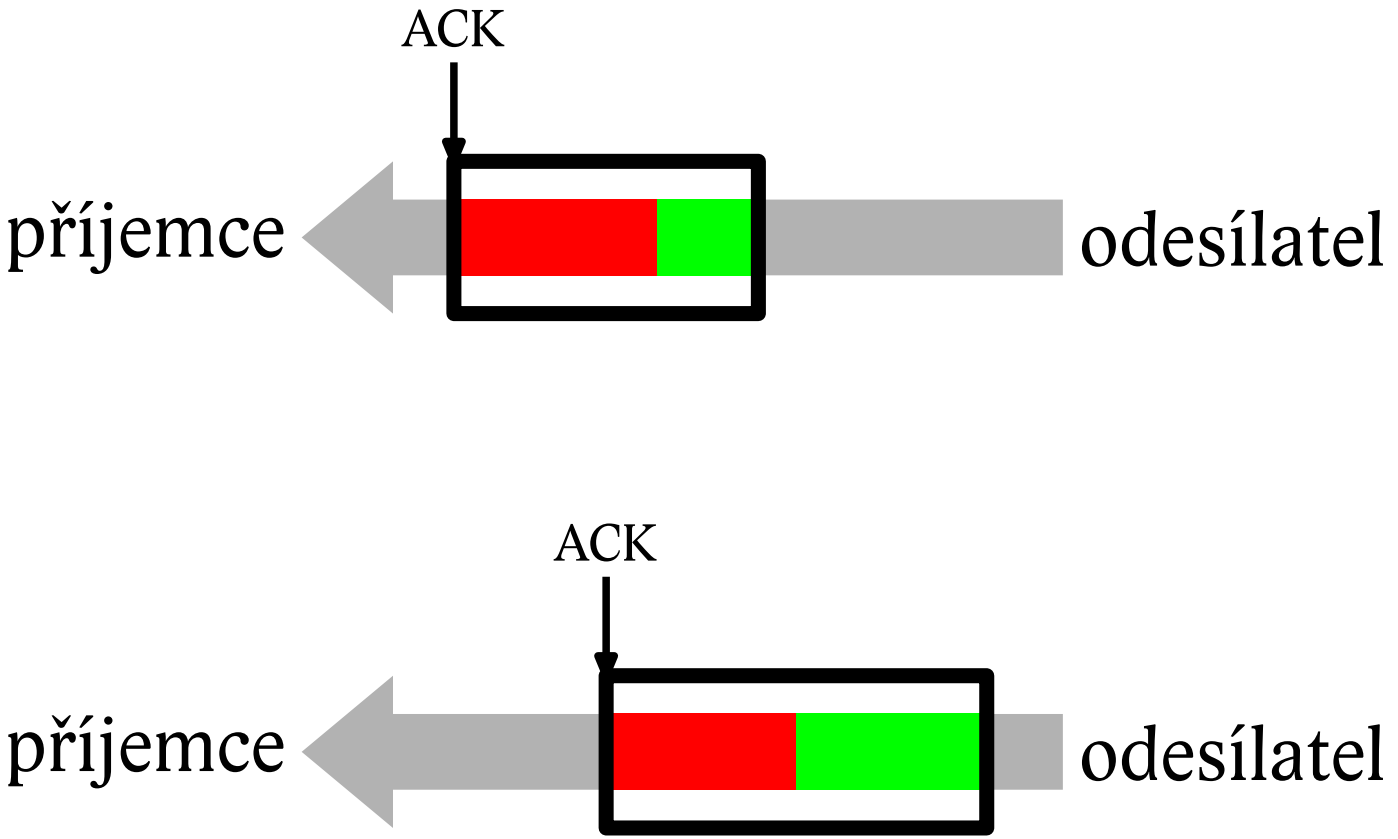
1. Server „poslouchá“ na určitém portu – čeká na zahájení spojení.
2. Klient posílá TCP segment s příznakem SYN, nastavuje sekvenční číslo, velikost okna a obvykle také max. velikost segmentu.
3. Server odpovídá segmentem SYN+ACK, potvrzuje (SN klienta + 1) a nastavuje vlastní parametry (SN, WS a volitelně MSS).
4. Klient odpovídá segmentem ACK, potvrzuje (SN serveru + 1).

Ukončení spojení

Protokol TCP: Ukončuje se zvlášť každý směr spojení, vyvolává buď klient nebo server. V praxi se ale většinou ukončí obě poloviny spojení bezprostředně za sebou.

1. Ukončující strana: segment s příznakem FIN
2. Druhá strana: segment s příznakem ACK

Klouzající okno



Pomalý start, kongesční okno

Co dělat, když je úzké místo spojení někde na trase?

Algoritmus navrhl Van Jacobson v 80. letech: rychlost odesílání segmentů se reguluje podle rychlosti přicházejících potvrzení (ACK).

Kongesční okno: jeho velikostí server sám limituje rychlost odesílání dat.

Slow start: Kongesční okno se nastaví na 1 segment. Za každý obdržený ACK se zvýší o 1 (max. do velikosti klouzajícího okna), než se některý segment ztratí. Tato velikost CWIN se označuje jako *práh kongesce (CT)*.

Algoritmus pak zmenší CWIN zpět na 1 a roste exponenciálně až do $CT/2$, potom však roste už jen lineárně.

Long Fat Pipes

$$\textit{capacity}\ [\text{bit}] = \textit{bandwidth}\ [\text{bit/s}] \times \textit{round-trip time}\ [\text{s}]$$

Spojení s velkou kapacitou limituje propustnost TCP spojení, často na zlomek šířky pásma.

Dvojí problém:

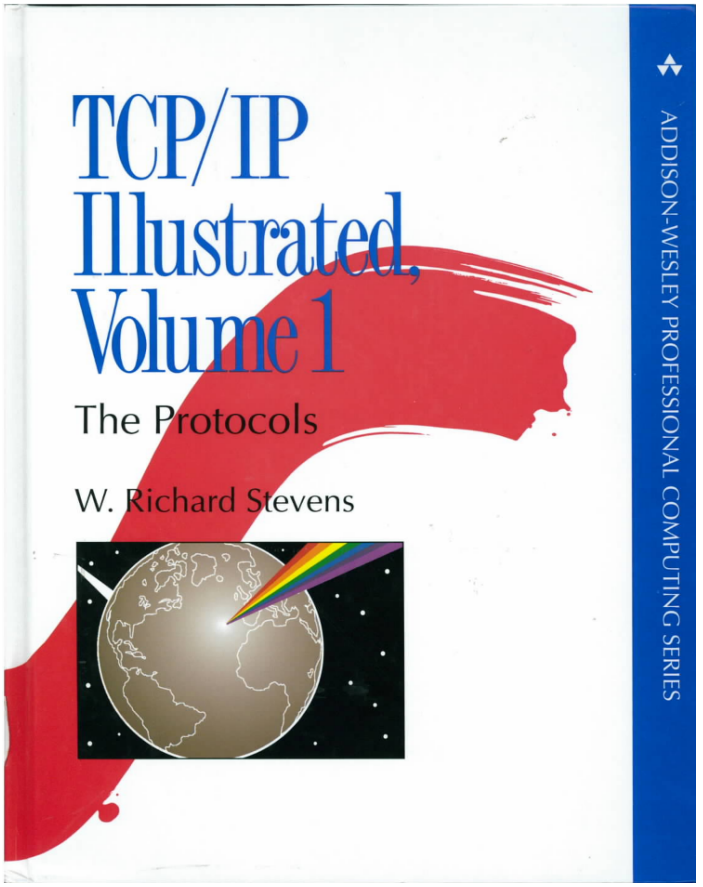
1. klouzající okno – jen 16 bitů. Řešení: Window Scale option
2. náhodné ztráty segmentů. Řešení: úpravy algoritmu

   Tuning: *http://www.psc.edu/networking/projects/tcptune/*

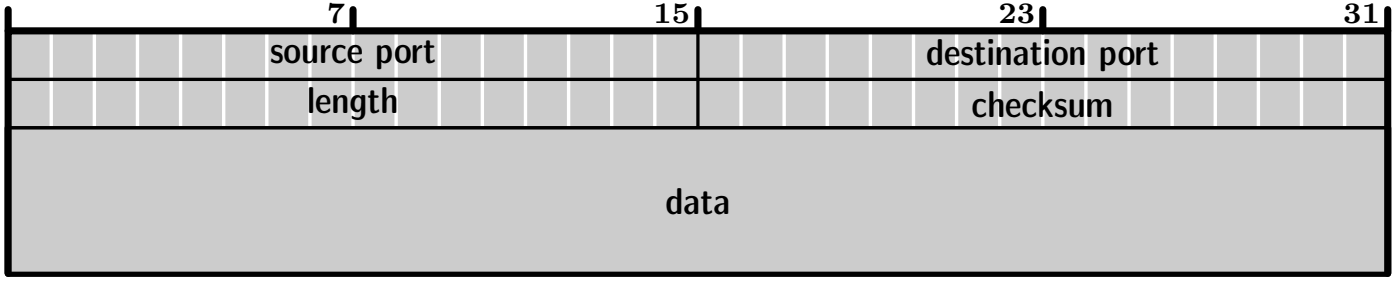
Útoky proti TCP

- *SYN Flood* (útok typu DoS)
- *Connection hijacking* využívá nedostatečné náhodnosti počátečních sekvenčních čísel.

Další čtení o TCP



User Datagram Protocol



Ukázky aplikací

1. HTTP – World Wide Web
2. DNS – převod jmen na IP adresy
3. XMPP – Instant messaging (Jabber)

HyperText Transfer Protocol

HTTP/1.1 – definovaný v RFC 2616 (1999)

Protokol typu klient-server, klient (browser) posílá požadavky webovému serveru, který je zpracuje a odešle odpověď – ve většině případů webovou stránku HTML nebo její část.

HTTP používá TCP, standardním portem je 80. Jde o *bezstavový* protokol, takže aplikace musí případné zachcení stavu řešit vlastními cestami (cookies).

Dva režimy spojení:

1. krátkodobá spojení – každá TCP relace obsahuje jeden požadavek klienta a jednu odpověď.
2. perzistentní spojení – vše v jedné TCP relaci, která se ukončí až po vypršení stanoveného časového intervalu.

DNS

Domain Name System – hierarchická databáze doménových jmen. Zajišťuje převod doménových jmen na IP adresy a obráceně, nalezení serverů pro mail a jiné služby atd.

Architektura typu klient-server, komunikace mezi klientem a serverem prostřednictvím UDP, mezi servery TCP. Port je obvykle 53.

Dva typy dotazů:

1. *rekurzivní* – dotazovaný server má za úkol nalézt kompletní odpověď na dotaz a předat ji tazateli;
2. *iterativní* – tazatel hledá odpověď postupně po složkách doménového jména

Jabber, XMPP

http://www.jabber.org, *http://www.xmpp.org*

XMPP je otevřený protokol a Jabber open-source program založený na XMPP.

Vlastnosti:

- Distribuovaná architektura typu klient-server, velký počet vzájemně komunikujících serverů
- Každý klient se připojuje ke svému „domácímu“ serveru
- Komunikace pomocí TCP, obvyklé porty 5222 (spojení klienta na server), 5223 (totéž přes SSL), 5269 (mezi servery)

Portscan

Běžná metoda „osahávání“ sítě. Zjišťuje, jaké porty jsou na daném souboru počítačů otevřené.

Motivy: zvědavost, příprava útoku; bezpečnostní audit, inventura počítačů v síti

Běžně dostupné programy: nmap, SuperScan

Postup

1. Útočník/auditor postupně otevírá TCP spojení na posloupnost portů.
2. Je-li odpověď:
    - SYN+ACK: port je otevřený (listening)
    - RST+ACK: port není otevřený
    - nic: otevírající segment byl zablokován (např. firewallem)

Program *nmap*

```
$ sudo nmap 172.29.2.1
Starting Nmap 4.53 ( http://insecure.org ) at 2008-09-23 15:40 CEST
Interesting ports on marconi.lhotkovi.cz (172.29.2.1):
Not shown: 1711 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
53/tcp open  domain
80/tcp open  http
MAC Address: 00:0F:66:92:51:DA (Cisco-Linksys)
Nmap done: 1 IP address (1 host up) scanned in 13.427 seconds
```

Časové charakteristiky

```
 $ traceroute -n www.google.com
traceroute to www.l.google.com (74.125.39.147), 30 hops max, 40 byte packets
 1  195.113.161.129 9.642 ms 9.219 ms 8.934 ms
 2  195.113.156.154 12.514 ms 12.825 ms 12.603 ms
 3  195.69.144.247 27.833 ms 28.339 ms 30.116 ms
 4  209.85.248.88 29.382 ms 37.101 ms 209.85.248.93 29.998 ms
 5  72.14.232.209 39.026 ms 209.85.254.100 29.671 ms 209.85.250.141 35.071 ms
 6  209.85.254.112 35.391 ms 38.203 ms 209.85.254.118 35.454 ms
 7  209.85.248.44 38.948 ms 43.601 ms 209.85.254.134 51.056 ms
 8  209.85.254.112 52.559 ms 74.125.39.147 35.911 ms 37.936 ms
```

Zpoždění a jitter

Zpoždění paketů (end-to-end delay) ovlivňuje kvalitu „služby“ (QoS), tj. objektivní nebo subjektivní parametry různých aplikací.

Důležité parametry:

- *Velikost zpoždění (latence)* – problém především pro interaktivní (real-time) aplikace: IP telefonie (≤ 250 ms), videokonference, síťové hry.
- *Rozptyl zpoždění (jitter)* – problém i pro jiné audiovizuální aplikace.

Složky zpoždění

$$d_{hop} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{hop} – celkové zpoždění na jednom „hopu“

d_{proc} – zdržení způsobené zpracováním paketu

d_{queue} – zdržení ve frontách směrovače

d_{trans} – zpoždění při vysílání

d_{trans} – zpoždění způsobené konečnou rychlostí šíření signálu

Metody sběru dat

1. Packet traces
2. Informace o datových tocích
3. Simple Network Management Protocol (SNMP)

Packet traces

Pro kompletní záznam provozu slouží řada programů označovaných jako *packet sniffers*: tcpdump, Wireshark, Kismet (WiFi), Ettercap (Ethernet), . . ., knihovna libpcap.

Provoz na vysokorychlostních linkách není dost dobře možné ukládat, ale je nutné ho průběžně zpracovávat nebo agregovat. Platí i v případě, že ukládáme pouze hlavičky.

Datové toky

RFC 3917: Datový tok (IP traffic flow) je množina IP paketů procházející přes *bod pozorování* (směrovač, sonda) během jistého *časového intervalu*. Všechny pakety patřící danému toku se shodují v určitém souboru parametrů, jimiž mohou být

1. jeden či více údajů z hlaviček IP (např. zdrojová nebo cílová IP adresa), transportní vrsty (např. číslo zdrojového nebo cílového portu) nebo aplikační vrstvy
2. jeden či více dalších parametrů paketu
3. jeden či více parametrů odvozených při zpracování paketu (next hop address, vstupní/výstupní síťové rozhraní)

Standardní soubor parametrů (klíč): zdrojová a cílová IP adresa, protokol (TCP, UDP, ICMP), zdrojový a cílový port.

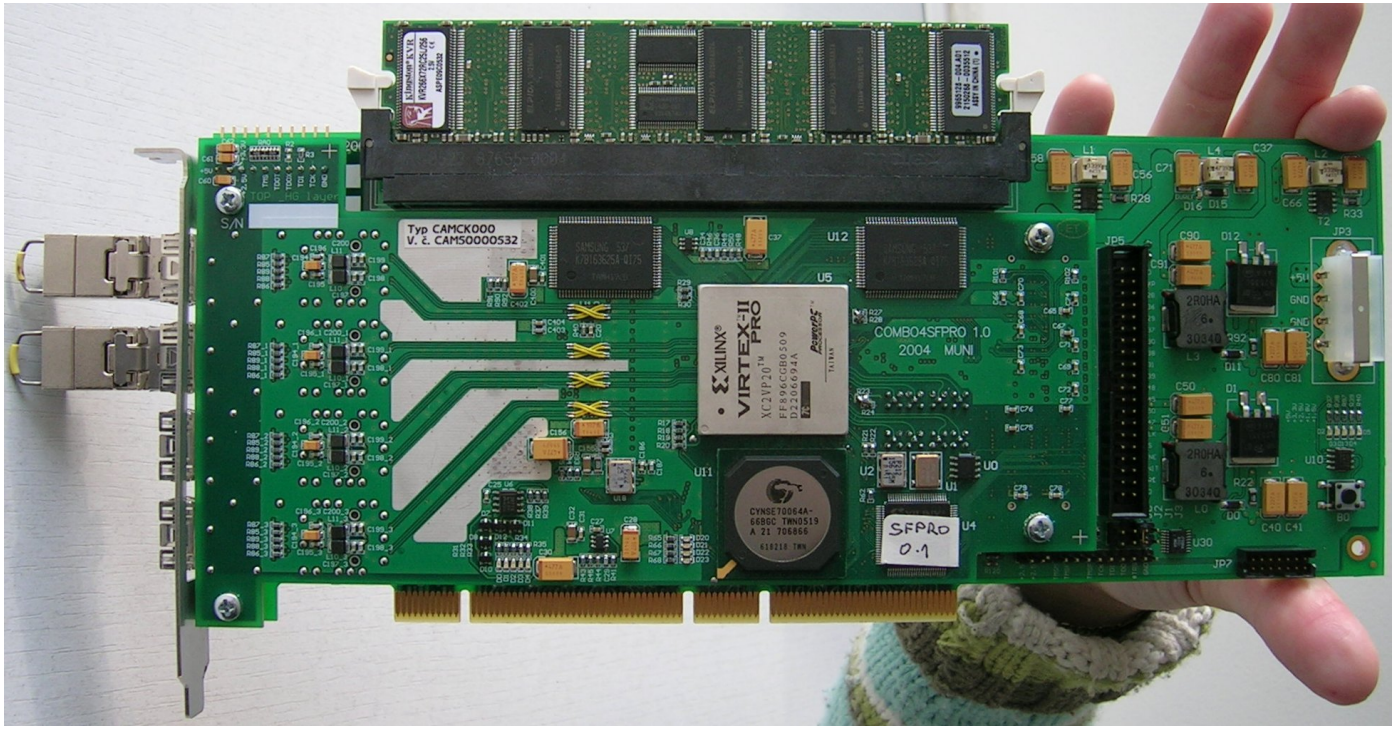
Jak získat informace o tocích?

Informace o tocích mohou exportovat některé (dražší) směrovače nebo specializované hardwarové či softwarové sondy.

Protokoly pro datový export:

1. NetFlow – proprietární protokol fy Cisco Systems
2. IPFIX – otevřený standard vyvíjený v rámci IETF [RFC 3917]

Sonda FlowMon



Internet Management Framework

Je typu klient-server, ale POZOR! Serverů je obvykle větší počet (*managed devices* – (PC, servery, tiskárny, směrovače, čidla), zatímco klient je jeden nebo dva (*management station*).

Součásti:

- *Network management objects* – co se monitoruje a co s tím může administrátor dělat?
- *Data modelling language* – v jaké formě se informace předávají?
- *Komunikační protokol* – jak se domluví klient se serverem?

Management objects

Každý objekt reprezentuje konkrétní údaj (konfigurační parametr, statistiku) nebo jejich posloupnost.

Soubor (databáze) objektů se nazývá *Management Information Base (MIB)*.

Objekty jsou dále tématicky členěny (podle typu zařízení nebo služby apod.) do MIB modulů.

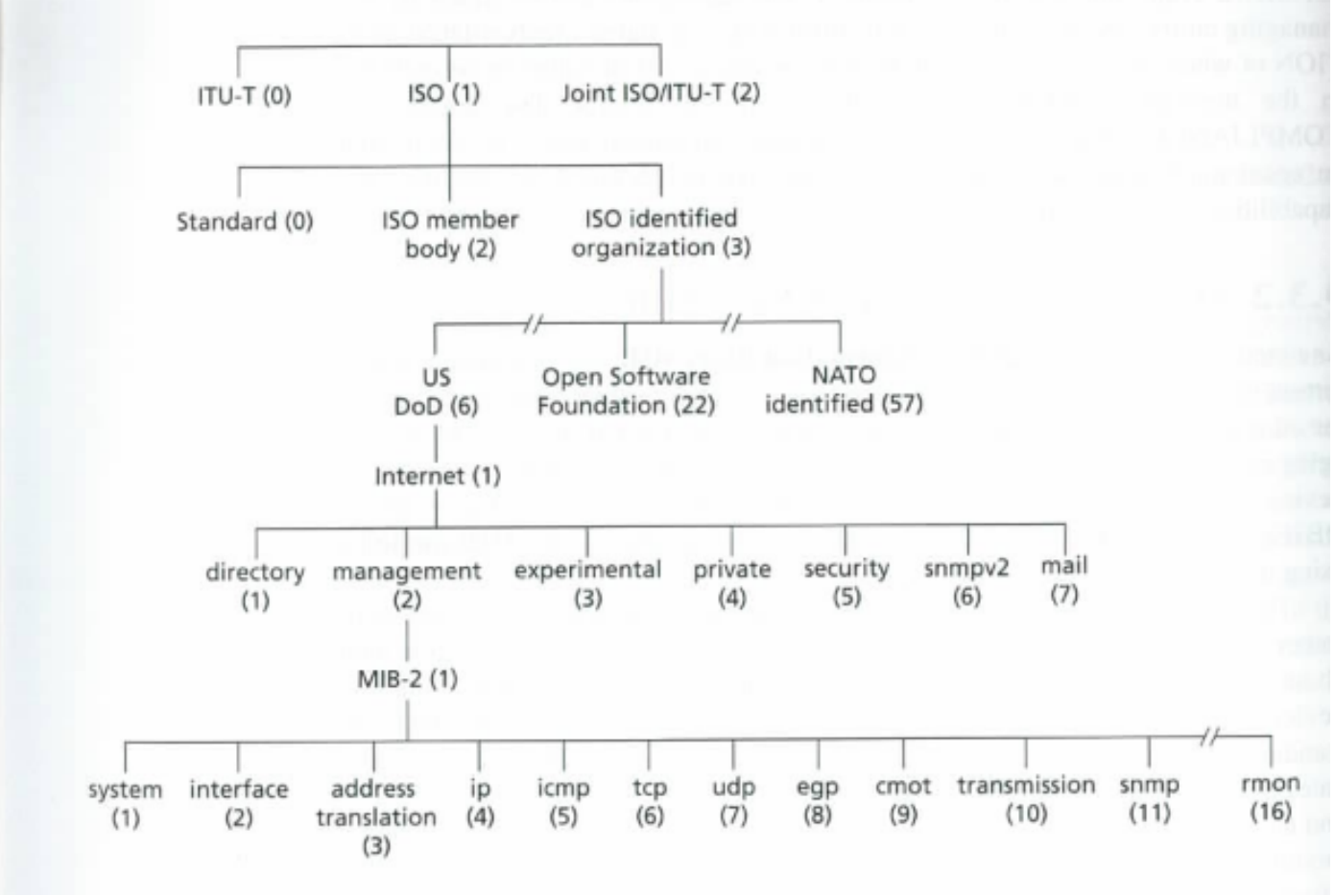
Data modelling language

SMI (Structure of Management Information): RFC 2578–2580

Datové typy: například integer32, unsigned32, counter32, counter64, gauge32, octet string, object identifier, timeTicks

```
ipInDelivers OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The total number of input datagrams
                    successfully delivered to IP
                    user-protocols (including ICMP)."
    :: {ip 9}
```

Struktura MIB



Komunikační protokol

SNMP (Simple Network Management Protocol): RFC 3416 (verze 2)

Na straně serveru (managed device) se o komunikaci stará *agent* (softwarový démon).

Zabezpečení přístupu: *SNMP communities* (read-only, read-write)

Dva režimy:

1. Dotaz-odpověď: klient/manažer požádá agenta o určitou akci, ten ji provede (nebo také ne) a pošle odpověď.
2. *Trap*: asynchronní zpráva generovaná agentem, která manažera upozorňuje na změnu stavu zařízení nebo nějaký jev.

Dotazy na data

U všech jsou jako parametry OID objektů, jejichž hodnota se žádá.

- `GetRequest` jedna nebo více hodnot MIB objektů
- `GetNextRequest` hodnota následujícího prvku posloupnosti
- `GetBulkData` umožňuje přenést celé posloupnosti

Software pro SNMP

- Net-SNMP:
  http://www.net-snmp.org
- MRTG (Multi-Router Traffic Grapher):
  http://oss.oetiker.ch/mrtg/
- Nagios:
  http://www.nagios.org/