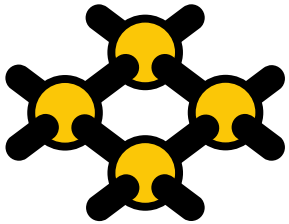
PB156cv – Aplikační vrstva (cvičení)

Lukáš Ručka

SITOLA FI MU

17. května 2021



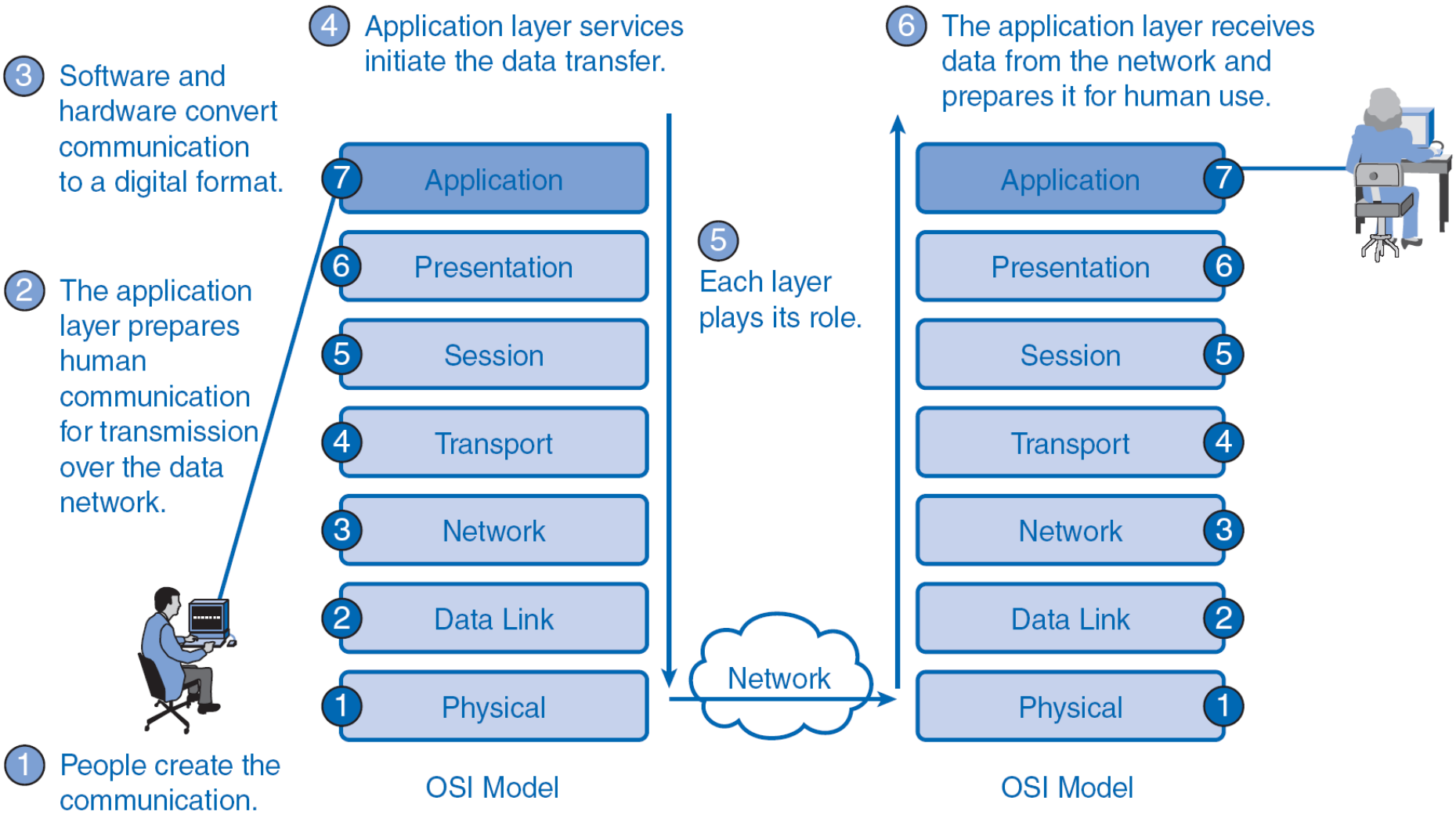
Laboratoř pokročilých síťových technologií

Cíle

- Seznámit se s některými principy návrhu protokolů aplikační vrstvy
- Porozumět tomu, jak fungují síťové aplikace a jaké jsou jejich možnosti
- Něco málo naprogramovat

Teorie a historie

Připomenutí vrstev



Co je to aplikační protokol?

- Zprávy, které si mezi sebou zasílají aplikace
  $\Rightarrow$ formát zpráv, jejich pořadí, odpovědi, chybové stavy...
- Neplést s API (I = rozhraní), protokol je o sémantice
- Z principu interaktivní

Standardizace

- Konsorcia a komunity, RFC
- Proprietární vs. otevřené
  $\Rightarrow$ například HTTP, FTP vs. Skype
- Vendor lock-in
- Známé, registrované a dynamické porty

Řízení

In-band vs. out-of-band signalizace

- Oddělené porty pro data a řízení
- Hlavičky vs. payload

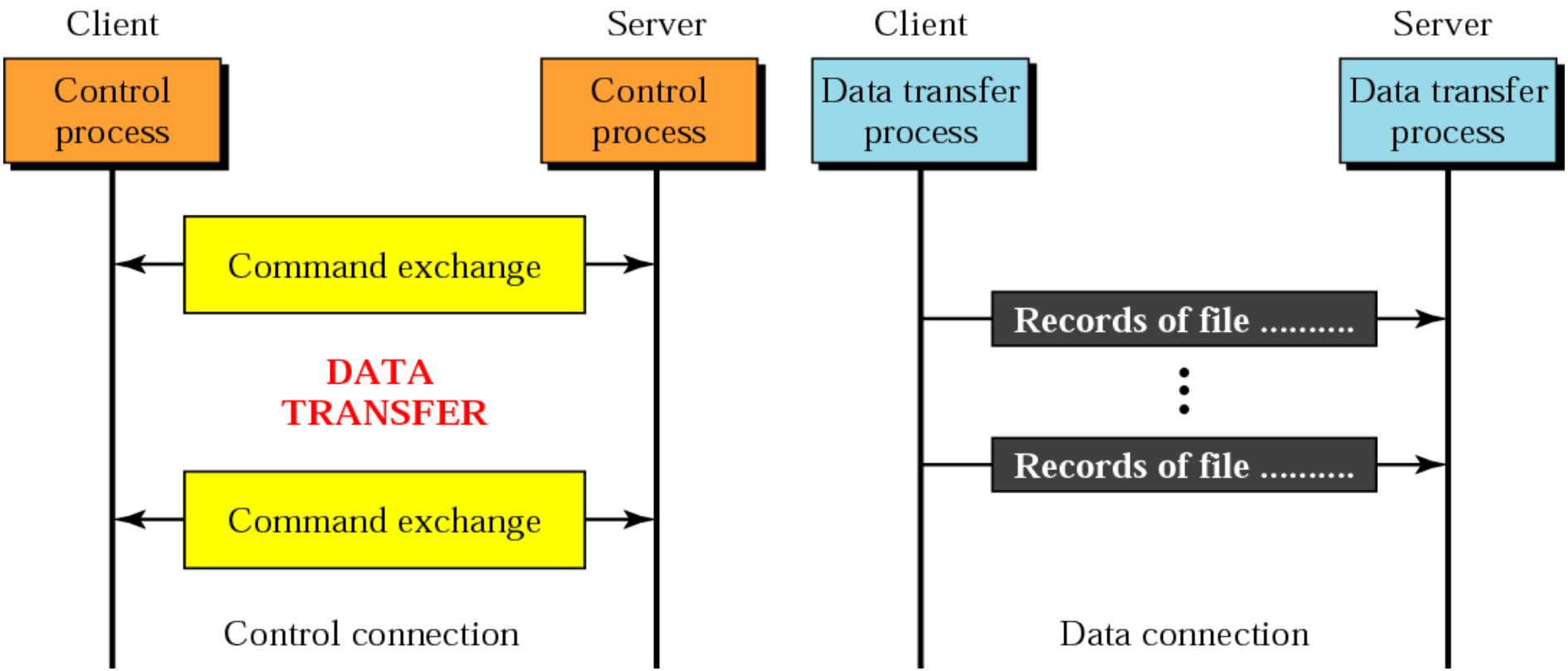
Multitasking

- Single-socket – např. UDP
- Multi-socket – např. TCP (ne nutně)

Specifikace protokolů

- Plaintext – textové popisky zpráv a jejich formátů
- Ilustrace
- Formálním automatem
- MSC

Hrubé MSC pro FTP



Příklady na konkrétních protokolech

FTP – File Transfer Protocol

- Textový protokol
- Klient se připojí na control port serveru (21) z portu (N), server (20) na data port klienta (N-1)
- RFC 959
- RFC 1579 zavedlo pasivní režim

FTP – Demo

- Připojit se na ftp.linux.cz, standardní řídící port (telnet)
- USER, PASS
- TYPE A
- LIST
- TYPE I
- EPSV
- RETR

SMTP a HTTP

SMTP

- TCP kanál, port 25, naprosto nedůvěryhodný
- Email se dělí na obálku a obsah
- Jde o prastarý protokol – součástí handshake je i dojednání endianity

(Demo)

HTTP

- TCP kanál, port 80
- Řízení založeno na hlavičkách, následovných tělem
- V pozdějších revizích přidána meta-hlavička Host
- Protokol je známý soupiskou stavových kódů (404 – Stránka nenalezena)
- Málo známé: jazykové preference prohlížeče

(Demo)

SDP – Session Description Protocol

```
v=0
o=- 0 0 IN IP4 127.0.0.1
s=No Name
t=0 0
a=tool:libavformat 58.17.101
m=video 5004 RTP/AVP 96
c=IN IP4 172.17.1.220
b=AS:200
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-id=1
m=audio 5006 RTP/AVP 97
c=IN IP4 172.17.1.220
b=AS:48
a=rtpmap:97 opus/48000/2
```

Práce se sockety – demo

TCP server

Fáze

1. Aktivace síťového stacku – socket
2. Provázání s portem (a adresou) – bind
3. Vytvoření fronty příchozích spojení – listen
4. Přijetí spojení – accept (+fork)
   1. Obsluha (read, write, select)
   2. Uvolnění pomocného socketu – close
5. Uvolnění primárního socketu – close

TCP server

Fáze

1. Aktivace síťového stacku – socket
2. Provázání s portem (a adresou) – bind
3. Vytvoření fronty příchozích spojení – listen
4. Přijetí spojení – accept (+fork)
   1. Obsluha (read, write, select)
   2. Uvolnění pomocného socketu – close
5. Uvolnění primárního socketu – close

TCP client

Fáze

1. Aktivace síťového stacku – socket
2. Zjištění IP adres cíle – getaddrinfo / gethostbyname
3. Spojení – connect
4. Obsluha (read, write, select)
5. Uvolnění socketu – close

UDP

Sockety otevřeny oběma směry

1. Aktivace síťového stacku – socket
2. Pokud odesíláme jako první, zjištění IP adres cíle
3. Provázání portu a socketu – bind
4. Střídání odesílání příjmu – sendto / recvfrom[1]
5. Uvolnění socketu – close

[1] `man 2 select`

Úkoly

Odevzdávání

- Úkoly budete odevzdávat jako samostatné soubory, zabalené do jediného archivu (.tar)
- Tento archiv budete odevzdávat odesláním na port 27703/tcp na hostu 147.251.54.177, sice způsobem uvedeným v patičce tohoto snímku.
- Archiv se vyhodnocuje jednou za den, v nočních hodinách. Můžete tedy postupně přidávat jednotlivé položky.
- K datům do archivu přiložte identifikační token, stažený zde. `https://fi.muni.cz/~xrucka/PB156cv/token.pb156cv`.
- Pokud není uvedeno jinak, platí pro danou část deadline 31. května.

```
cat l7hw.tar | socat -  \
tcp4-connect:147.251.54.177:27703
```

Úkol 1: RFC a reference na protokoly

Dohledejte RFC dle klíče níže a podejte report:

- Přidělené RFC dle $(\prod_{digit \in uco} digit) \mod 8500$
- Text mema uložte ve formátu TXT do souboru memo.txt
- Report o memu ve formátu json dle níže uvedného klíče uložte do souboru memo.json
- Do souboru nonip.txt uložte stručný report o Vámi vybraném síťovém protokolu, svým využitím zhruba odpovídajícímu aplikační vrstvě; včetně odkazu na doprovodné informace o něm.[2]

```
{ "rfc": "1234", "doi": "125.456/77a8", "status":
"experimental", "name": "HTTP 2.0 text mode" }
```

[2] Předpokládána samostatná práce, nikoliv 5 protokolů které odevzdají všichni.

Úkol 2: Rešerše

Vyhledejte všeobecně 1 zajímavost ze světa počítačových sítí. Nejlépe takovou, která souvisí či nějak jinak dopadá na aplikační vrstvu.

Zajímavost uložte do textového souboru interesting.txt, který přibalíte do archivu.

Úkol 3: TCP/UDP chat

Ve studijních materiálech najdete rozpracované zdrojové kódy chatovacího klienta. Tohoto klienta dokončete se zohledněním znalostí nabytých z pondělní hodiny.

Pro úlohu budou dvě sady testů - první se bude spouštět každý den, až do 31. května. Reportem z této sady bude, jestli vám program funguje na 100%, nebo zda je na něm potřeba ještě zapracovat. Výstupem druhé sady pak bude nápověda, co je ještě rozbyté. Deadline těchto testů 7. června. 2. sada testů bude spouštěna náhodně a bude i přispívat i do discordu, kde můžete zahlédnout i nápovědy jak svou implementaci opravit.

Ti, kdo odevzdají plně funkční řešení již na první pokus nemusí odevzdávat úkol č. 2.

Úkol 4: ICMP jako MTU discovery protokol

- Velikost ethernetového rámce se může mezi různými L2 sítěmi lišit.
- MTU na cestě zjistitelné snadno – nejvyšší velikostí nefragmentovaného packetu (hint: ping).[3]
- Změřte MTU ke stroji 147.251.54.177 a zjištěnou hodnotu uložte jako jediný řádek do souboru MTU.txt, který odevzdáte jako součást archivu. Stroj je nakonfigurován tak, aby měnil MTU každou 10. minutu.

```
2020-04-20 19:30:54 1492
```

[3] V ideálním světě by vám při nesprávném MTU přišla ICMP zpráva fragmentation needed. Z technických důvodů toto v našem setupu nelze zajistit a tak se musíte řídit největším packetem, na který ještě dostanete odpověď.

Bonus

Soupiska užitečných síťových nástrojů

- nmap
- netstat / sockstat (ss)
- tc
- ip
- ping
- nc / netcat
- telnet
- wget, curl
- ns-3 / ns-4
- socat
- links, links2, elinks, lynx
- netem
- dig
- host
- mail / mailx
- ipcalc
- mininet
- iptables, ebtables, ufw
- pf (BSD), authpf (BSD)