

MUNI
ÚVT

PB156 – cvičení
Aplikační vrstva a její použití

Tomáš Rebok & Tamara Šejnová
Ústav výpočetní techniky MU
Fakulta informatiky MU

Představení vyučujících

RNDr. Tomáš Rebok, Ph.D. (FI MU, ÚVT MU)

- senior výzkumný pracovník, vedoucí projektů a různých skupin
- dlouhodobá činnost v oblasti (pokročilých) počítačových sítí a jejich aplikací (již od dob dávných studií 😊)
- dlouhodobá činnost v oblasti náročných výpočtů a zpracování dat



Tamara Šejnová (ÚVT MU)

- studentka
- členka síťového týmu specialistů na ÚVT MU



Cíle dnešního cvičení

Uvědomění si vlastností aplikační vrstvy

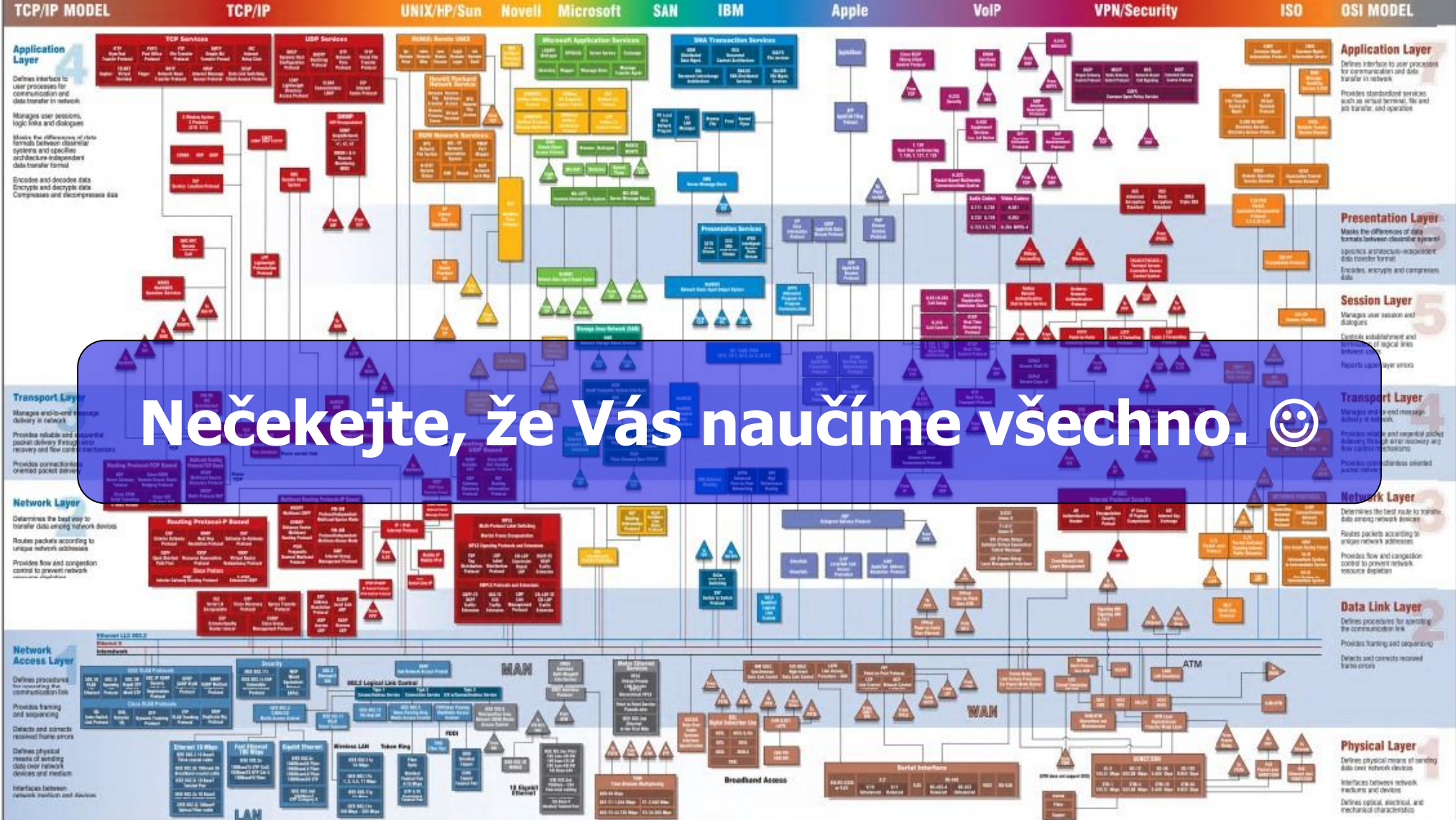
Seznámení s metodami/nástroji pro analýzu a správu vlastností sítě

Seznámení se základními protokoly aplikační vrstvy a jejich analýzou

Základní informace o síťových simulátorech

Programování síťových aplikací

Network Protocols Map



Nečekejte, že Vás naučíme všechno. 😊

L7 – Aplikační vrstva

Aplikační vrstva I.

Nižší vrstvy poskytují komunikační kanál mezi (většinou dvěma) komunikujícími stranami

- **UDP** kanál – analogie „poštovního doručovatele s balíčky“
- **TCP** kanál – analogie „roury“
- **porty** – identifikace konkrétní aplikace (na jednom serveru jich je mnoho)

Aplikační vrstva II.

L7 zahrnuje síťové aplikace/programy a aplikační protokoly:

- **aplikační protokoly** (HTTP, SMTP, atd.) jsou součástí síťových aplikací (web, email)
 - nejedná se o aplikace samotné
 - protokoly definují formu komunikace mezi komunikujícími aplikacemi
- **aplikační protokoly** definují:
 - typy zpráv, které si aplikace předávají (*request/response*)
 - syntaxi přenášených zpráv
 - sémantiku přenášených zpráv (jednotlivých polí)
 - pravidla, kdy a jak aplikace zprávy vysílají

Aplikační vrstva III.

Způsob komunikace (struktura a sémantika zasílaných zpráv) je zcela na aplikacích (programátorovi)

- aplikační vrstva nevynucuje žádný komunikační protokol
 - syntaxe i sémantika je čistě v režii aplikace (programátora)
- zabezpečení může být realizováno jak na nižších vrstvách („zabezpečená roura“, SSL/TLS), tak na úrovni vlastní aplikace (v režii programátora)

Užitečné síťové nástroje/aplikace

aneb

„Nejen analýzou paketů živ je síťář...“

Užitečné síťové nástroje

Úkol: Zjistěte maximální MTU cesty k serveru

- např. server: caine.ics.muni.cz
- **odpověď**: použijeme **ping**
 - *Úkol a)* Proč dokážeme zaslat maximálně 1472 B?
 - *Úkol b)* Která zpráva je generována jako reakce na příliš velký paket?
 - zachyťte síťový provoz, kde si své tvrzení doložíte
 - další užitečné možnosti ping:
 - flood ping: `ping -f ...`
umožňuje diagnostiku ztrátovosti paketů a problémů s průtokem v síti

Užitečné síťové nástroje

Ovlivnění vlastností síťových linek/přenosů

- Jaké vlastnosti/parametry na síťových linkách/přenosech rozeznáváme?
 - šířka pásma, ztrátovost, latence, jitter, duplikace paketů, poškození paketů, atp.
- proč můžeme chtít parametry uměle měnit (zhoršovat)?
 - testování chování aplikací v různých situacích (analogie reálných podmínek)

Úkoly: Na odchozí lince svého stroje nastavte:

- a) 10% **poškození** paketů
- b) 20% **duplikaci** paketů
- c) 10% **ztrátovost** a 500ms **latenci** paketů současně

- **odpověď:** použijeme **netem**
- d) Jak všechny nastavené parametry smažeme?
- e) Jak netem parametry nastavíme na příchozích linkách?

Užitečné síťové nástroje

Testování vlastností síťových linek/přenosů

- Jakým nástrojem/nástroji dané parametry/vlastnosti otestujeme?
- **odpověď**: použijeme **iperf**, **ping**, atp. (znáte z předchozích sem.)

- a co v případě potřeby **testování specifického datového provozu**?
 - tj. potřebujete zachovat přesně definovanou syntaxi i sémantiku datových paketů
 - napíšeme si vlastní 😊
 - Úkol: **Popište, jak byste k danému problému přistoupili...**

Užitečné síťové nástroje

Analýza otevřených portů vzdáleného stroje

- **odpověď**: použijeme **nmap** (znáte z předchozích cvičení)

Analýza otevřených portů lokálního stroje

- a zjištění, které aplikace/procesy tyto porty otevřely?
- **Úkol**: Zanalyzujte otevřené porty Vašeho stroje a minimálně u 3 z nich popište, které procesy a aplikace je otevřely?
 - **odpověď**: použijeme **netstat**
 - jak bude podle Vás vypadat typické volání `netstat` ?

Užitečné síťové nástroje

Připomenutí z minulých cvičení

- **iperf** – testování přenosové rychlosti sítě
 - UDP i TCP provoz, half/full-duplex, různé velikosti, ...
- **traceroute** – zjištění uzlů (směrovačů) po cestě k cíli
 - analýza s využitím postupného zvyšování TTL
- **ethtool** – konfigurace vlastností síťových rozhraní
 - zjišťování vlastností a změna některých parametrů
- **ip** – konfigurace síťových rozhraní
 - modernější alternativa k nástrojům ifconfig, route, arp, brctl, atp.

Jmenná služba (DNS)

Užitečné síťové nástroje

Domény a doménová jména, IP adresy

Co je to doména a (plné) doménové jméno uzlu v síti?

- a) Jak zjistíte plné doménové jméno lokálního stroje?
 - **odpověď**: použijeme **hostname/domainname**
- b) Jak zjistíte IP adresu ke známému doménovému jménu a naopak?
 - **odpověď**: použijeme **host**

Zjištění informací o konkrétní doméně

- c) Jak zjistíte podrobné informace (datum registrace, kontakty, atp.) o konkrétní doméně?
 - **odpověď**: použijeme **whois**

Zjištění informací k podsíti

- d) Jak zjistíte základní informace o IP adresách ke konkrétní podsíti?
 - **odpověď**: použijeme **ipcalc**

Užitečné síťové nástroje

Domény a doménová jména, IP adresy

Úkol: **Podrobně zanalyzujte DNS záznamy jednoho vybraného webového serveru a popište důležité informace jeho domény.**

Na lokální síti můžete chtít zavést jmennou službu pro snazší orientaci mezi servery.

- Jak si postavit vlastní jmennou službu pro privátní stroje?
 - **odpověď**: použijeme **/etc/hosts**
 - pokud potřebujeme pouze pro SSH přístupy, může být výhodnější **~/.ssh/config**
 - Úkol: **Nakonfigurujte si vlastní pojmenování vybraných strojů/serverů (využijte oba přístupy) a otestujte chování nástroje ,host‘ pro DNS resolving.**
 - můžete také zachytit pakety zasílané v rámci DNS resolvingu a zanalyzovat je

Webové služby (HTTP)

Užitečné síťové nástroje

Webové služby

Které nástroje standardně využíváme pro přístup k webovým službám?

- webové prohlížeče
- ty vás učit nebudeme 😊
 - vyjma zmínění **links**: textově orientovaný webový prohlížeč
 - alternativy **links2**, **lynx**, ...
 - Kdy může být textový prohlížeč užitečný?

wget – nástroj pro stahování souborů přes HTTP, HTTPS i FTP

- jednoduché stahování i pokračování v přerušném stahování
 - včetně stahování pouze v případě, kdy se zdroj změnil
- zrcadlení celých webů a rekurzivní stahování
 - rekurze = zpracování odkazů již stažených stránek a další stahování
 - včetně přizpůsobení odkazů pro lokální prohlížení

Užitečné síťové nástroje

Webové služby

cURL – velmi mocný nástroj na stahování, ale i nahrávání dat

- podporuje celou řadu protokolů
 - HTTP, HTTPS, FTP, FTPS, LDAP, IMAP, POP3, SMTP, ...

Pro stažení jedné stránky můžete použít příkaz:

```
$ curl -O http://www.apple.com/downloads/
```

Pro upload dokumentů:

```
$ curl -T data.sit http://mujucet@idisk.mac.com/Documents/
```

Užitečné síťové nástroje

Webové služby

curl – podpora při psaní REST aplikací (další možnost „HTTPIe“)

- odeslání POST requestu bez jakýchkoliv dat
`curl -X POST http://URL/example.php`
- odeslání POST requestu s daty
`curl -d "data=example1&data2=example2" http://URL/example.cgi`
- odeslání POST requestu s formulářovými daty
`curl -X POST -F "name=user" -F "password=test" http://URL/example.php`
- odeslání POST requestu s obrázkem v příloze
`curl -X POST -F "image=@/path/example.gif" http://URL/uploadform.cgi`
- odeslání POST requestu s JSON dokumentem
`curl -H "Content-Type: application/json" -X POST -d
'{"user":"bob","pass":"123"}' http://URL/`
- pro více informací: „curl -help“ nebo „curl -manual“

Elektronická pošta

Užitečné síťové nástroje

Elektronická pošta

Tradičně realizovaná protokoly:

- **SMTP** – odesílání pošty
- **POP3/IMAP** – příjem/čtení pošty

Simple Mail Transfer Protocol (SMTP)

- jedna z nejstarších aplikací v prostředí internetu
- zpráva se skládá z **hlavičky** a **těla zprávy**
 - tělo může obsahovat (kromě vlastní textové zprávy) také volitelné přílohy s libovolným obsahem
 - s využitím tzv. MIME hlaviček, do kterých je netextový obsah zakódován

Užitečné síťové nástroje

Elektronická pošta

Ukázka komunikace s poštovním serverem (C – zpráva klienta, S – odpověď serveru):

- **\$ telnet SERVER 25**
 - **S:** 220 mail.example.com ESMTP Postfix
 - **C:** EHLO example.net
 - **S:** 250 Hello example.net
 - **C:** MAIL FROM: <sender@example.net>
 - **S:** 250 Ok
 - **C:** RCPT TO: <friend@example.com>
 - **S:** 250 Ok
 - **C:** DATA
 - **S:** 354 End data with <CR><LF>.<CR><LF>
 - **C:** From: sender@example.net
 - **C:** To: friend@example.com
 - **C:** Subject: test message
 - **C:**
 - **C:** Hello!
 - **C:** .
 - **S:** 250 Ok: queued as 12345
 - **C:** QUIT
 - **S:** 221 Bye

Užitečné síťové nástroje

Elektronická pošta

Úkol: programem telnet se připojte na poštovní server MU (relay.muni.cz, port 25) a odešlete si emailovou zprávu.

▪ **Upozornění: Následující úkol je určen pouze pro testovací a výukové účely! Jakékoliv zneužití je zakázáno!**

- uvědomte si, že adresa odesílatele za hlavičkou „From: “ **nemusí být identická** jako adresa odesílatele v části „MAIL FROM: “
 - doručení probíhá podle adres na vnější obálce
 - příjemce emailu pak zobrazuje informace podle vnitřních hlaviček
- **Úkol: Na svou emailovou adresu si zašlete email s podvrženou (jinou) hlavičkou „From: “ (odlišnou od hlavičky „MAIL FROM:“)**
 - připojení na relay.muni.cz musí proběhnout z vnitřní sítě MU
 - pro svou vlastní bezpečnost si uvědomte, jak snadno lze podvrhnout email

Datové přenosy

Užitečné síťové nástroje

Datové přenosy na počítačových sítích

FTP – tradiční přístup k přenosům souborů na počítačové síti

- při používání běžného FTP jsou přihlašovací údaje přenášeny v textové podobě a je technicky možné je odchytit
- existuje **bezpečnější alternativa, tzv. FTPS**
 - zpětně kompatibilní s běžným FTP

SCP/SFTP – přenos souborů s využitím SSH protokolu

- vhodné v prostředí linuxových uzlů se spuštěným SSH serverem
- šifrovaný a poměrně rychlý přenos

Užitečné síťové nástroje

Datové přenosy na počítačových sítích

Nástroj NetCat – nc

- nešifrované zasílání dat mezi dvěma uzly s minimální reží
 - přímo do vytvořeného TCP kanálu
 - bez autentizace
- **příjemce** (musí být spuštěn první, lze využít libovolný port):
 - `$ nc -l -p 7555 > myfile.txt`
- **odesílatel** (stejný port jako příjemce):
 - `$ nc PŘÍJEMCE 7555 < myfile.txt`

Úkol: Vyzkoušejte si přenos souboru s využitím ,nc‘ a zanalyzujte zachycený datový provoz.

Bonusový úkol: Proměřte rychlosti přenosu všech tří variant (FTPS, SFTP, nc) na malých (stovky kB), středních (stovky MB) a velkých souborech (>10 GB)

Testování síťových aplikací

Užitečné síťové nástroje

Testování síťových aplikací

Jak testovat síťové aplikace v komplexních scénářích?

- **netem** sice umožní definici vlastností linek, ale nezajistí vybudování komplexnější síťové infrastruktury (topologie propojených uzlů)

Odpověď:

- použijeme **specializované síťové simulátory** (viz dále)
 - v případě, kdy nepotřebujeme pracovat s komplexními systémy
- alternativně **specializovaná prostředí poskytující flexibilní síťovou infrastrukturu**
 - většinou postavená nad virtualizovanými prostředími (cloudy)
 - např. Kybernetický Polygon (KYPO, bezpečnostní scénáře) vyvinutý na ÚVT MU

Užitečné síťové nástroje

Testování síťových aplikací

Mininet – <http://mininet.org>

- prostředí na bázi Linuxu umožňující vytvářet komplexní virtuální síť tvořenou simulovanými Linuxovými stanicemi a virtuálním síťovým přepínačem
 - **Open vSwitch** (<https://www.openvswitch.org>) – používaný síťový přepínač s komplexní funkcionalitou
 - lze definovat různé vlastnosti síťových linek
 - velmi oblíbený, určeno však pouze pro simulace drátových sítí

NS-3 - <https://www.nsnam.org>

- alternativní simulátor s diskretními událostmi
- umožňuje širší škálu protokolů, včetně simulace např. mobilních sítí
- existuje i ve variantě NS-4 (<https://github.com/P4Simulator>)

OpenNet (<https://github.com/dlinknctu/OpenNet>, pro SDN)

EstiNet (<http://www.estinet.com/ns>)

atp.

Užitečné síťové nástroje

Testování síťových aplikací

Bonusový úkol: **Vytvořte si jednoduchou simulovanou síťovou infrastrukturu a otestujte její základní vlastnosti.**

- v libovolném simulátoru
- nastavte různé parametry síťových linek
 - a tyto poté měřením prověřte

L7 – Pohled programátora

L7 – Pohled programátora

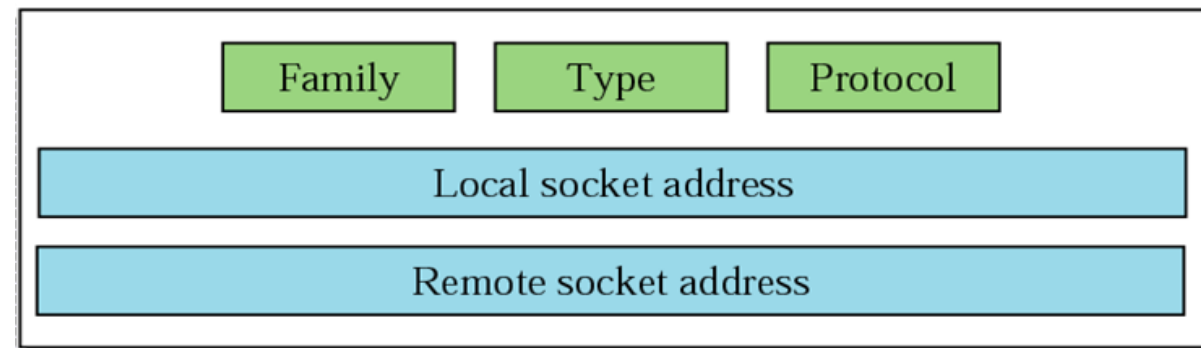
Aplikace komunikují skrze tzv. *sockety*

- struktura, která jednoznačně popisuje komunikující stranu (aplikaci)
- jsou nezbytné na obou komunikujících stranách

socket definuje:

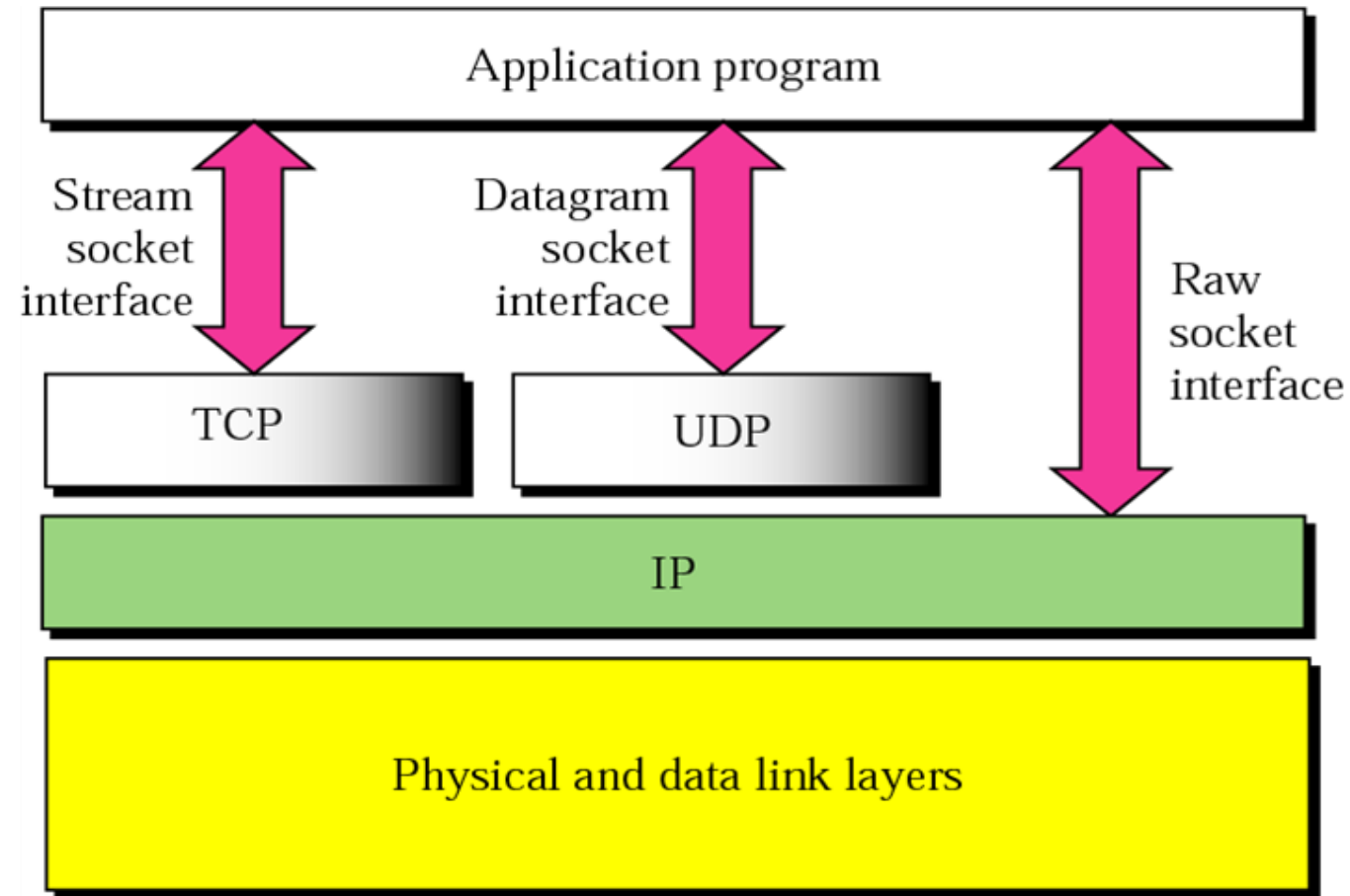
- **Family** – třída využitého protokolu (IPv4, IPv6, ...)
- **Type** – proudový (stream, TCP), datagramový (UDP) či základní (raw, ICMP)
- **Protocol** – obvykle nastaveno na 0 pro TCP a UDP
- **Lokální adresa socketu** – kombinace lokální IP a čísla portu lokální aplikace
- **Vzdálená adresa socketu** – kombinace IP a čísla portu vzdálené aplikace (komunikující strany)

Socket



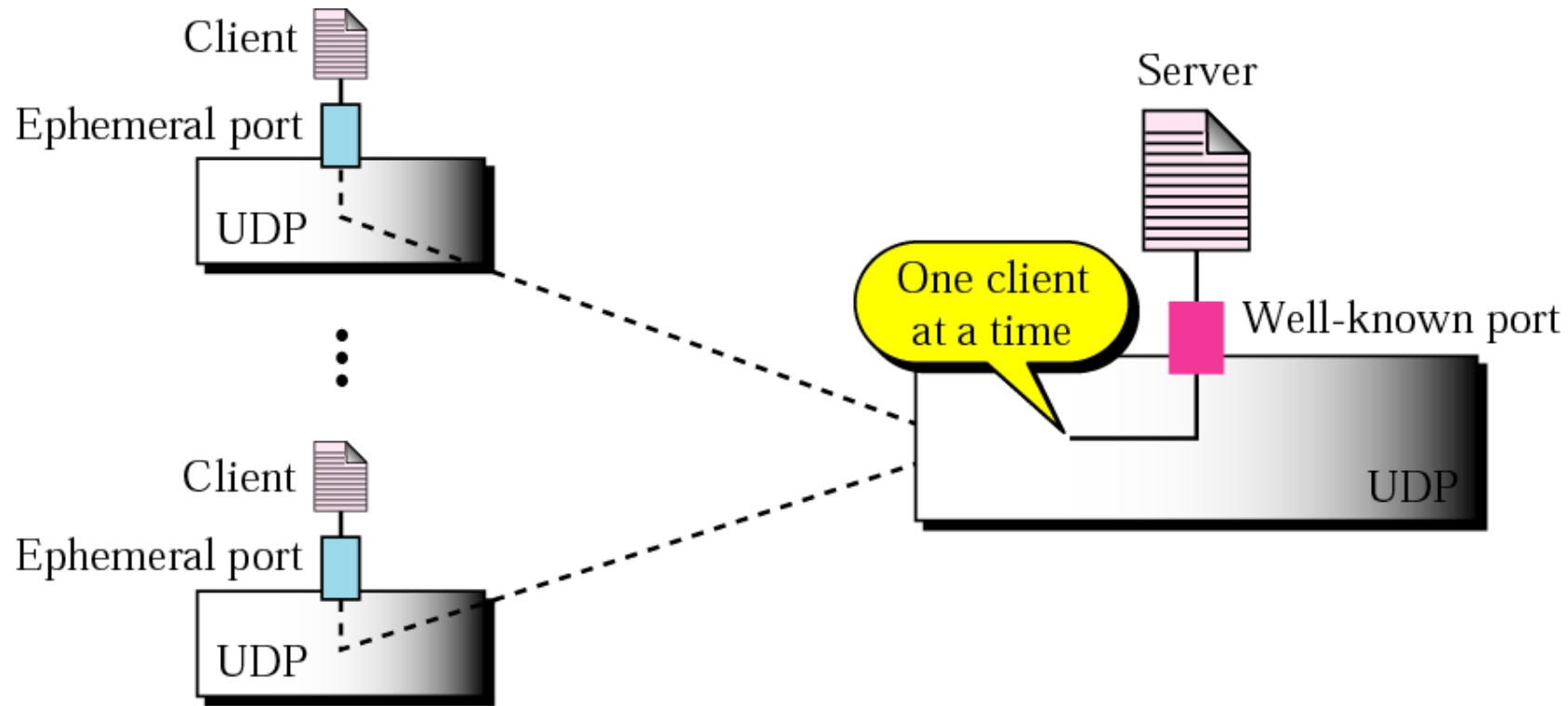
L7 – Pohled programátora

Typy socketů



L7 – Pohled programátora

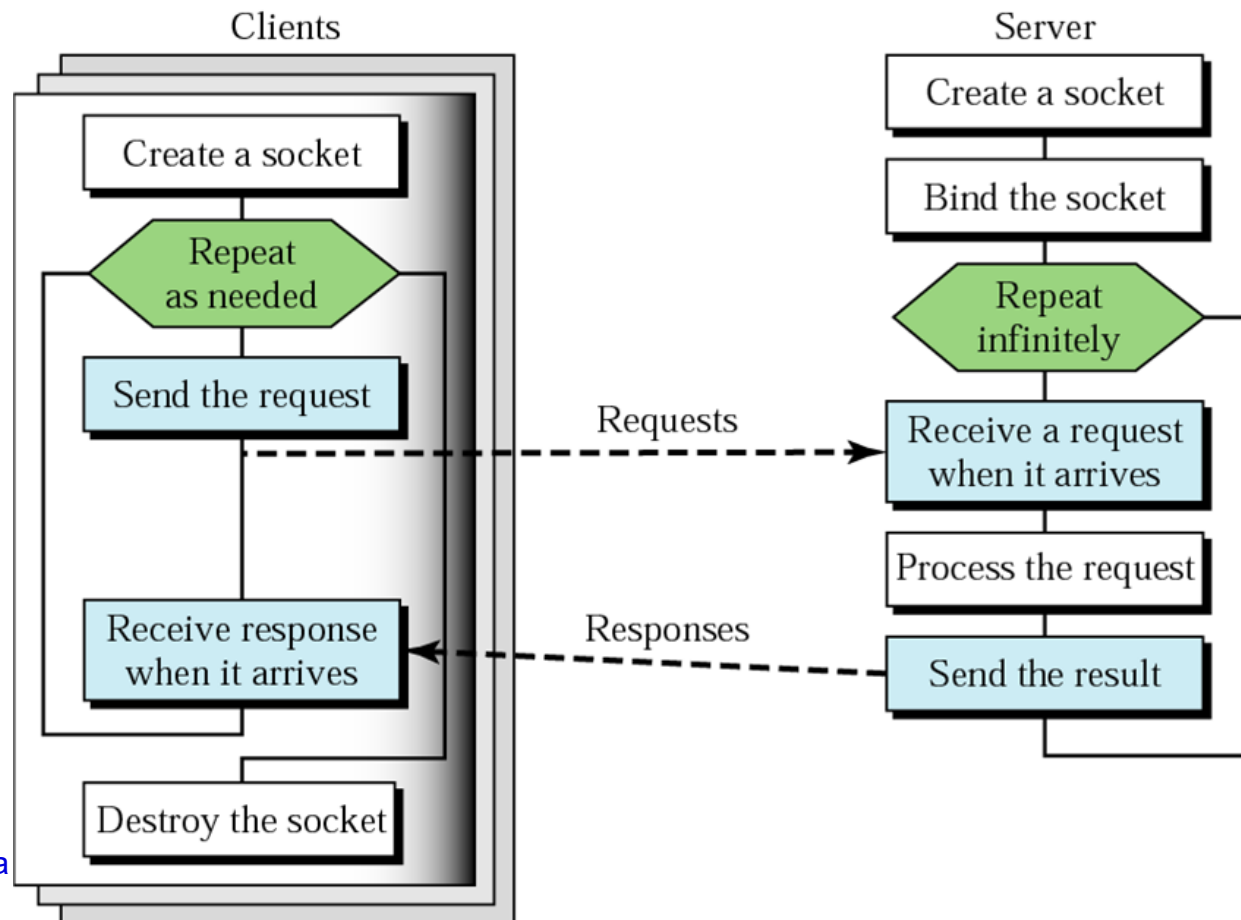
Obsluha socketu pro nespojovanou službu (UDP aplikace) I.



L7 – Pohled programátora

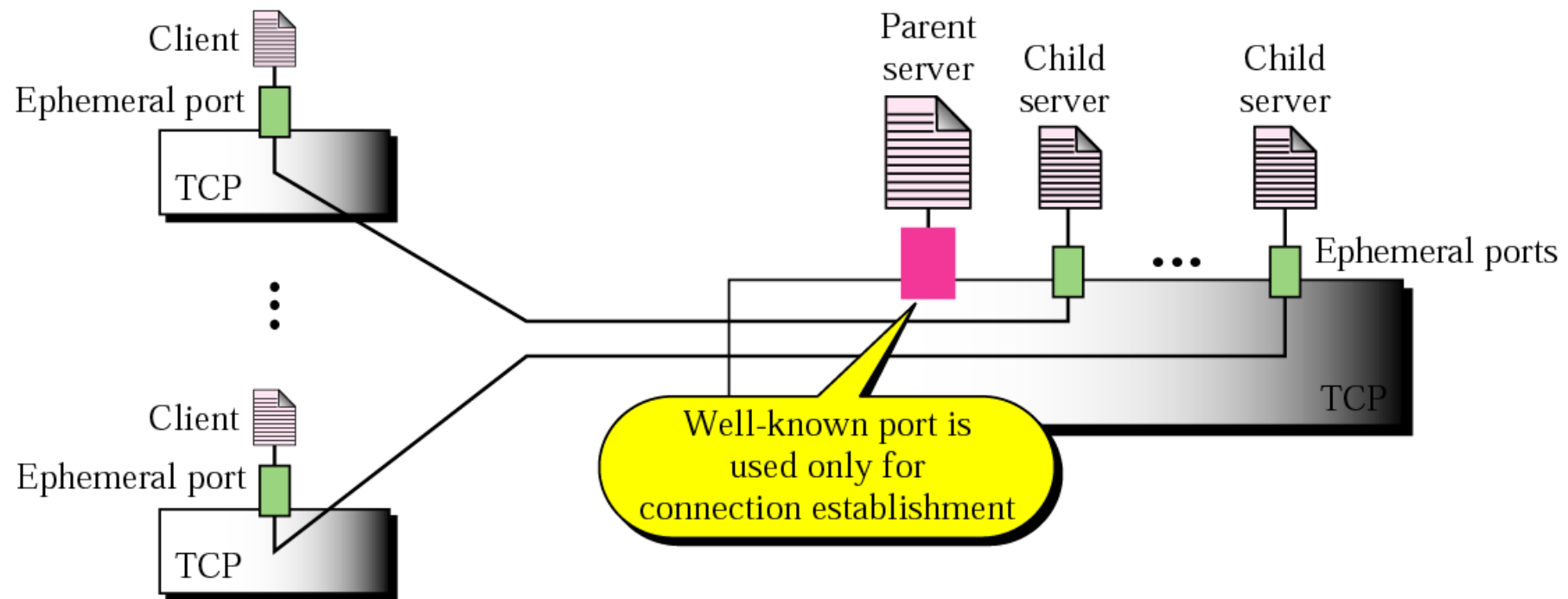
Obsluha socketu pro nespojovanou službu (UDP aplikace) II.

Each server serves many clients but handles one request at a time.



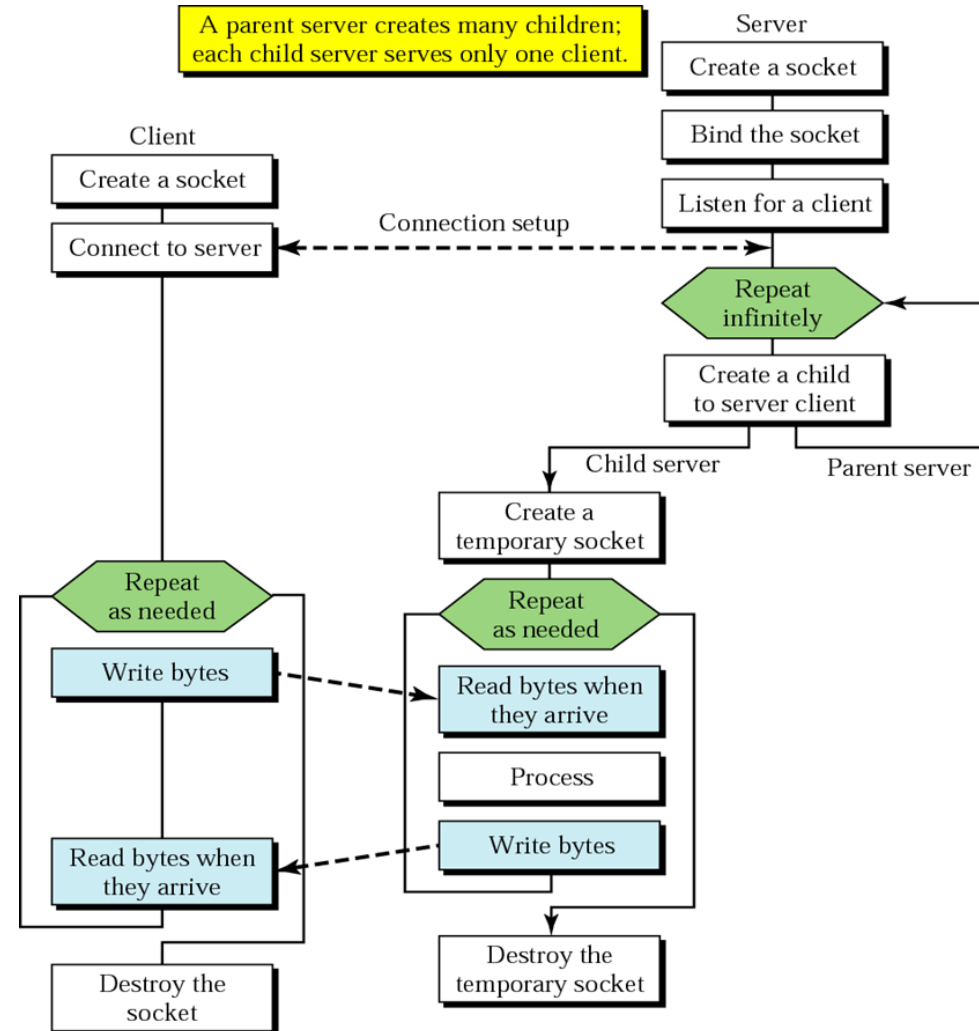
L7 – Pohled programátora

Obsluha socketu pro spojovanou službu (TCP aplikace) I.



L7 – Pohled programátora

Obsluha socketu pro spojovanou službu (TCP aplikace) II.



Programování síťových aplikací **chatovací aplikace**

Úkol: UDP aplikace

Jednoduchý chatovací klient

Server IP: 147.251.54.161

Port: 10000

Protokol:

- textové zprávy zaslané na UDP socket
 - prosím, buďte slušní! 😊

Úkol: UDP aplikace

Návod – seznam užitečných funkcí (Python3)

Knihovna „socket“

- a její import

Funkce „socket.socket()“

- otevření socketu

Funkce „socket.sendto()“

- zaslání dat

Poznámka: Lze programovat i v libovolném jiném programovacím jazyce.

cca 11 řádků

Úkol: TCP aplikace

Jednoduchý chatovací klient

Server IP: 147.251.54.161

Port: 10002

Protokol:

- textové zprávy zaslané na TCP socket
 - prosím, buďte slušní! 😊

Úkol: TCP aplikace

Návod – seznam užitečných funkcí (Python3)

Knihovna „socket“

- a její import

Funkce „socket.socket()“

- otevření socketu

Funkce „socket.connect()“

- navázání spojení

Funkce „socket.send()“

- zaslání dat

Funkce „socket.close()“

- ukončení spojení

Poznámka: Lze programovat i v libovolném jiném programovacím jazyce.

cca 13 řádků

Protokol

Protokol

Každý samostatně zpracuje a odevzdá protokol.

Protokol musí obsahovat minimálně následující části:

- **stručný popis řešení všech dílčích úkolů** realizovaných v rámci cvičení
 - (všechny úkoly uvozené „Úkol“)
 - zahrnuje použitý příkaz (včetně případného relevantních možností) a/nebo krátký textový popis
- **ke každé ze sekcí jednu zajímavost**, kterou naleznete na Internetu
- **jednu libovolnou zajímavost** ze světa počítačových sítí
 - primárně v oblasti síťových aplikací

MUNI
ÚVT