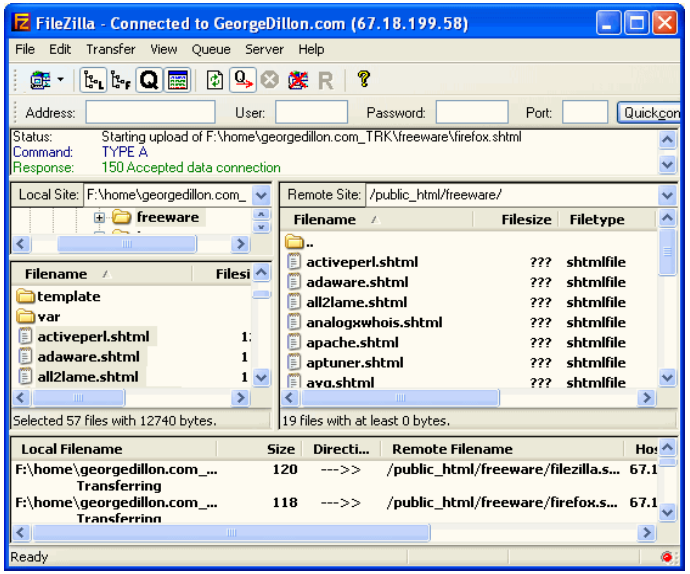
3. prednáška

Aplikačná vrstva

2.časť



DNS: Domain Name System

Ľudské identifikátory:

- Meno, RČ, č. OP, č. pasu

Koncové stanice, routre:

- IP adresa – používaná na adresovanie datagramov na sieťovej vrstve
- "doménové meno", napr. ics.upjs.sk – používané ľuďmi

DNS: preklad z doménových mien na IP adresy a naopak

Čo je DNS?:

- distribuovaná databáza nad celosvetovou hierarchiou mnohých doménových serverov (DNS serverov, name serverov)
- protokol aplikačnej vrstvy umožňujúci koncovým zariadeniam, routrom a DNS serverom komunikovať, aby umožnili preklad doménových mien a IP adries
  - poznámka: adresácia počítačov je záležitosť jadra internetu a predsa je v prípade doménových mien riadená z koncových zariadení

DNS

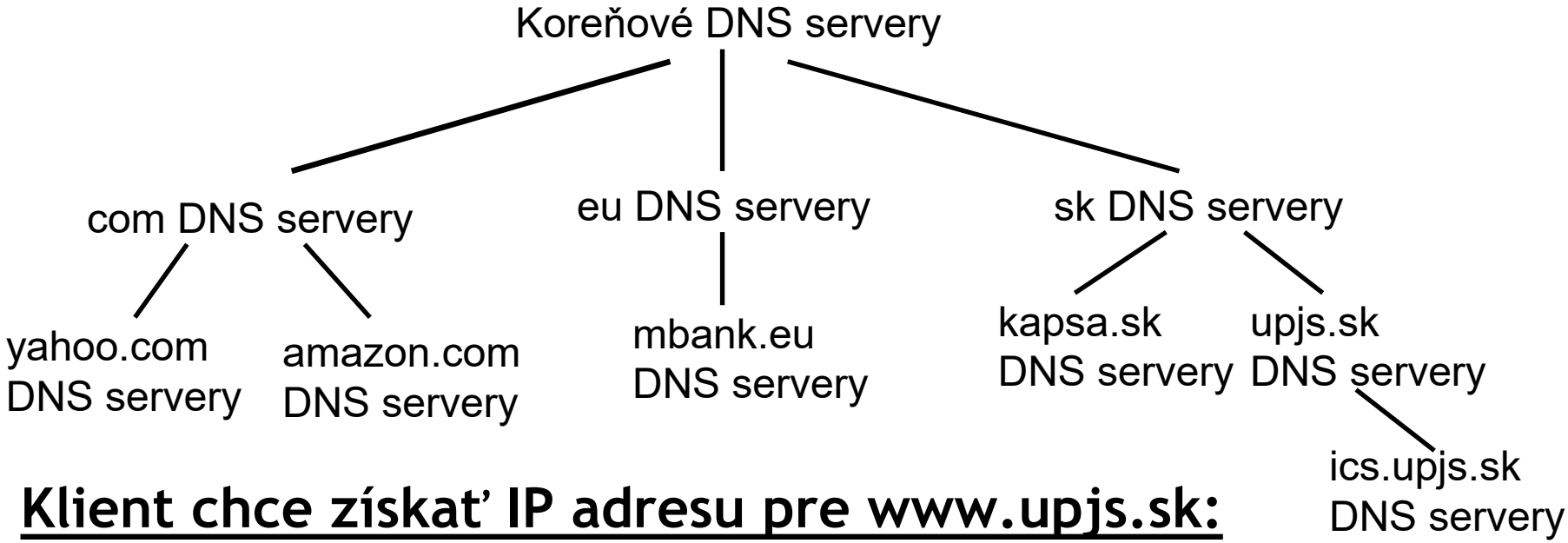
Služby DNS

- preklad doménových mien na IP adresy a späť
- host aliasing
  - Viac mien pre jeden počítač (jednu IP adresu)
- mail server aliasing
  - nezávislé od mena počítača
- distribuovanie záťaže
  - replikované webové servery: viac IP adries pre jedno doménové meno

Prečo nie je DNS centralizované?

- single point of failure
  - ak by centrálne DNS zlyhalo, "skolabuje" celý Internet
- veľká záťaž siete na jednom mieste
- vzdialenosť centrálnej databázy
- spravovanie

Distribuovaná, hierarchická databáza



Klient chce získať IP adresu pre www.upjs.sk:

- klient sa opýta koreňového DNS servera, aby mu povedal, kde sídli DNS server pre doménu sk
- klient sa opýta DNS servera pre doménu sk, aby mu povedal, kde sídli DNS server pre doménu upjs.sk
- klient sa opýta DNS servera pre doménu upjs.sk, aby mu povedal IP adresu pre www.upjs.sk

DNS: Koreňové doménové servery

- Kontaktované lokálnym doménovým serverom, ak nevie sám odpovedať
- Na svete je presne 13 IP adries, na ktorých sú koreňové DNS servery
  - Hovoríme, že je 13 koreňových doménových serverov
  - Reálne takmer každý z nich sídli na mnohých miestach
- Veľa počítačov s rovnakou IP adresou (viď. anycast)
- http://www.root-servers.org/

TLD a autoritatívne servery

- Top-level domain (TLD) servery:
  - Zodpovedné za com, org, net, edu, atď. A za všetky koncovky krajín sk, cz, at, uk, fr, ca, jp, ...
  - Doménu **com** spravuje Network Solutions
  - Doménu **sk** spravuje štátna spoločnosť SK-NIC
- Autoritatívne DNS servery:
  - DNS servery organizácií (napr. doménový server pre upjs.sk)
  - spravujú mená počítačov pre danú doménu
  - ich IP sú evidované v nadradených DNS serveroch (napr. TLD)
  - opak, teda neautoritatívne servery neexistujú
- existujú neautoritatívne odpovede od DNS serverov, ktoré si pamätajú preklad daného mena a IP adresy z predchádzajúcej komunikácie vo svojej cache

Lokálne doménové servery

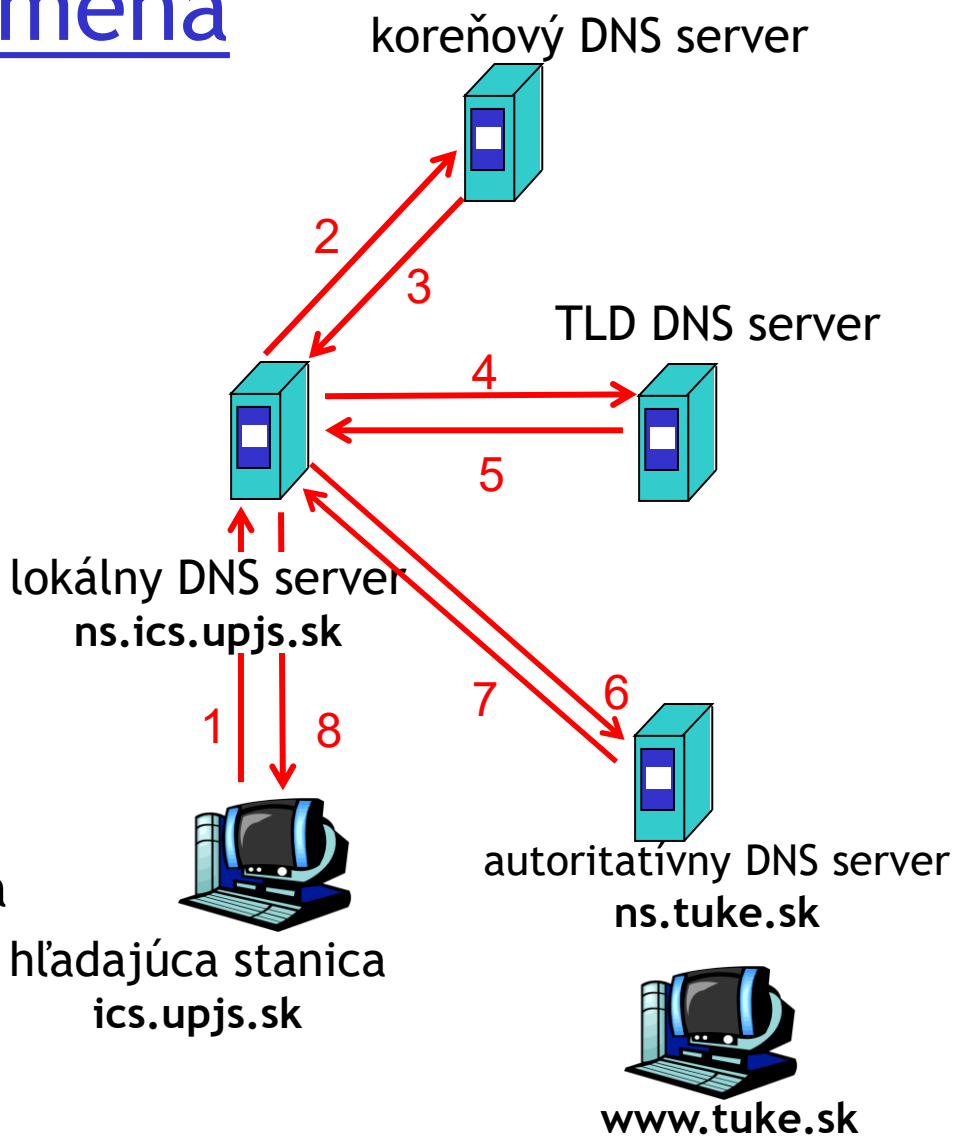
- nemusia byť zakomponované v hierarchii
- každý provider nejaký lokálny doménový server prevádzkuje
  - označuje sa ako “default name server”
- keď klient zadá požiadavku pre DNS, tento server komunikuje s ostatnými doménovými servermi a nakoniec vráti výsledok
  - niečo podobné ako proxy web server

Príklad prekladu mena na IP adresu

- Proces na ics.upjs.sk chce IP adresu pre www.tuke.sk

Postupné otázky:

- oslovený server odpovie menom servera, ktorý treba osloviť ako ďalší
- "Neviem odpovedať, spýtaj sa tohto servera"
- "Ja ho nemám, ja ho nemám, má ho číslo 195.12.159.3"



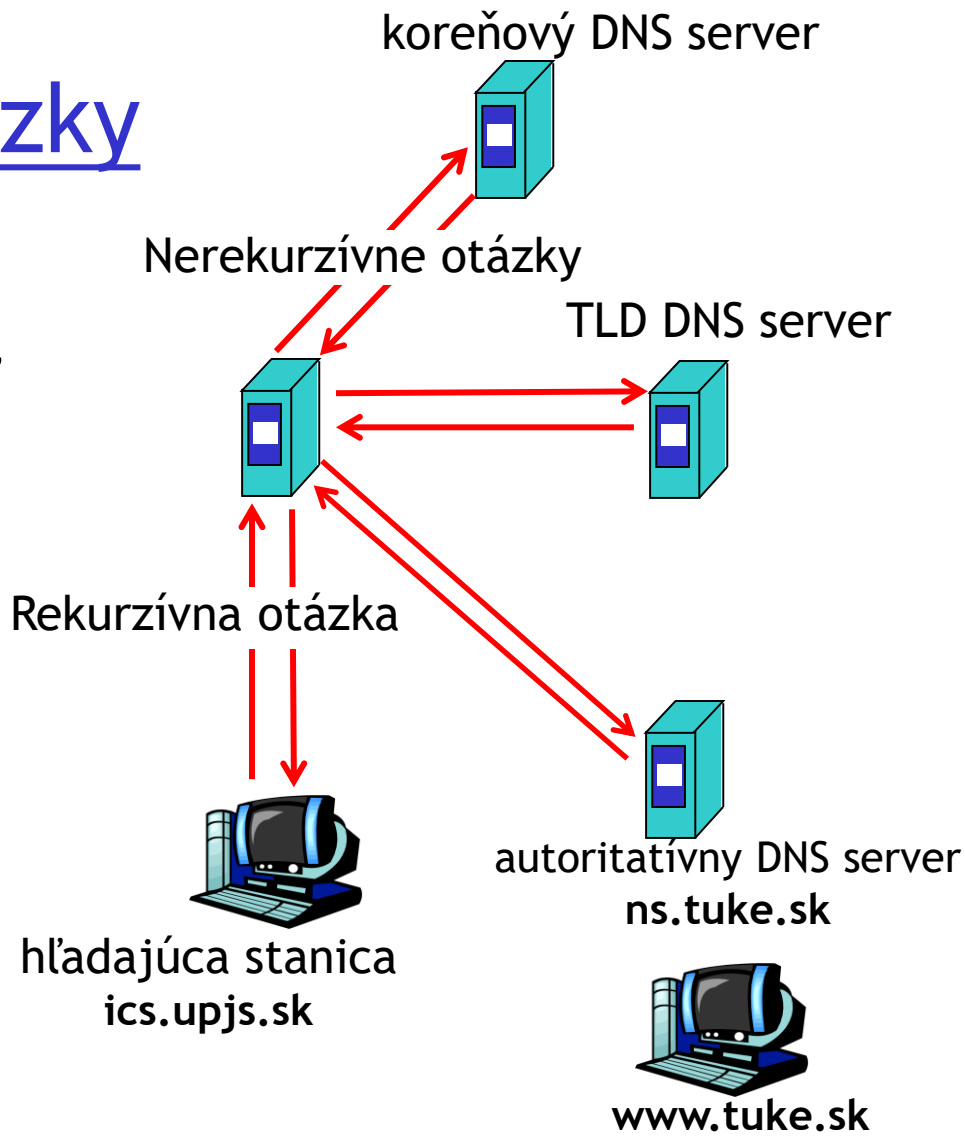
Rekurzívne a nerekurzívne otázky

Rekurzívne otázky:

- oslovený server zistí odpoveď sám a odpovie nám

Nerekurzívne otázky:

- oslovený server odpovie iba informáciami, ktoré pozná
- ak nevie odpovedať priamo, odpovie informáciami o serveroch, ktoré by mali vedieť bližšie informácie vedúce k odpovedi



DNS: cache-ovanie a obnovovanie záznamov

- keď sa nejaký server dozvie mapovanie z iného servera, tak si ho zapamätá v *cache*
  - záznamom v cache po čase vyprší platnosť
  - typicky blízke TLD servery sú známe lokálnym doménovým serverom
    - Teda sa nemusia zakaždým pýtať koreňových DNS serverov, kde sú TLD DNS servery

DNS záznamy: RR (resource records)

DNS: distribuované úložisko DNS záznamov

Formát RR: **(meno, hodnota, typ, ttl)**

Time To Live
=
dĺžka života

- Typ A
  - **meno** je doménové meno
  - **hodnota** je IPv4 adresa
- Typ AAAA
  - **to isté pre IPv6 adresy**
- Typ NS
  - **meno** je doména (napr. upjs.sk)
  - **hodnota** je meno autoritatívneho servera pre túto doménu
- Typ CNAME
  - **meno** je alias pre "kánonické" (skutočné) meno

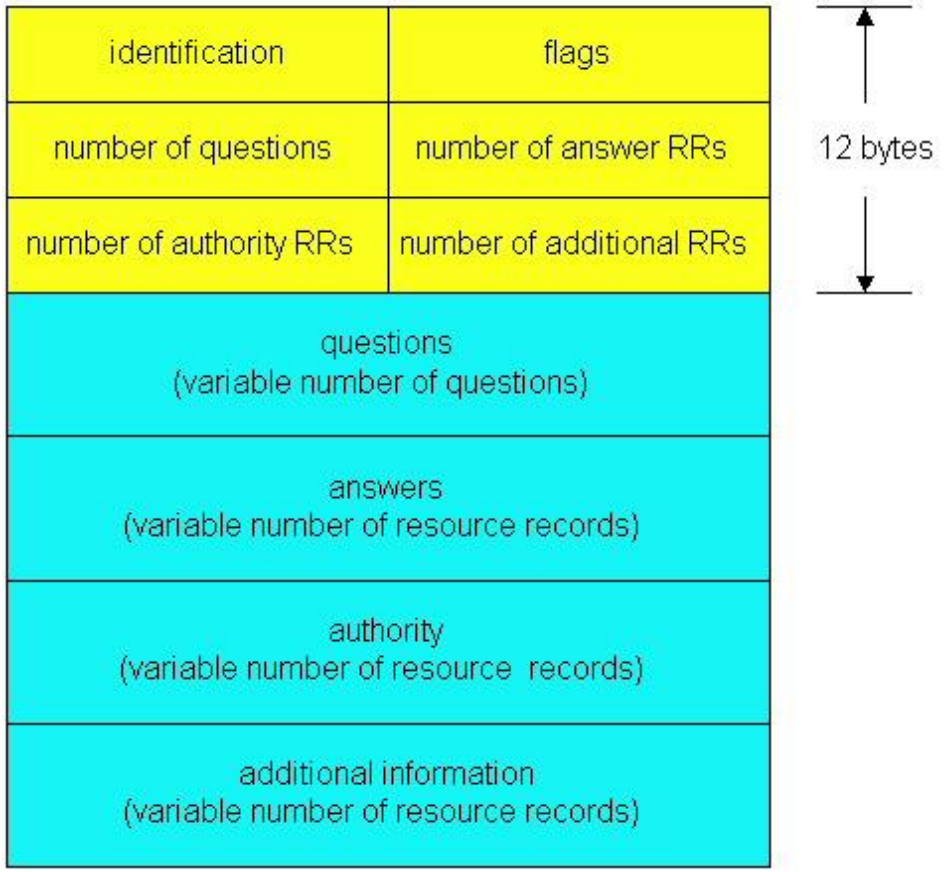
    www.ibm.com je v skutočnosti servereast.backup2.ibm.com
  - **hodnota** je kánonické meno
- Typ MX
  - **hodnota** je meno mailového servera asociovaného s **menom**

DNS protokol: RFC 1035

Požiadavky aj *odpovede* majú rovnaký formát – binárny

Hlavička správy:

- identification: náhodné 16 bitové číslo spoločné pre otázku a odpoveď
- flags (príznaky):
  - požiadavka alebo odpoveď
  - žiadosť o rekurziu
  - možnosť rekurzie
  - autoritatívna odpoveď alebo nie



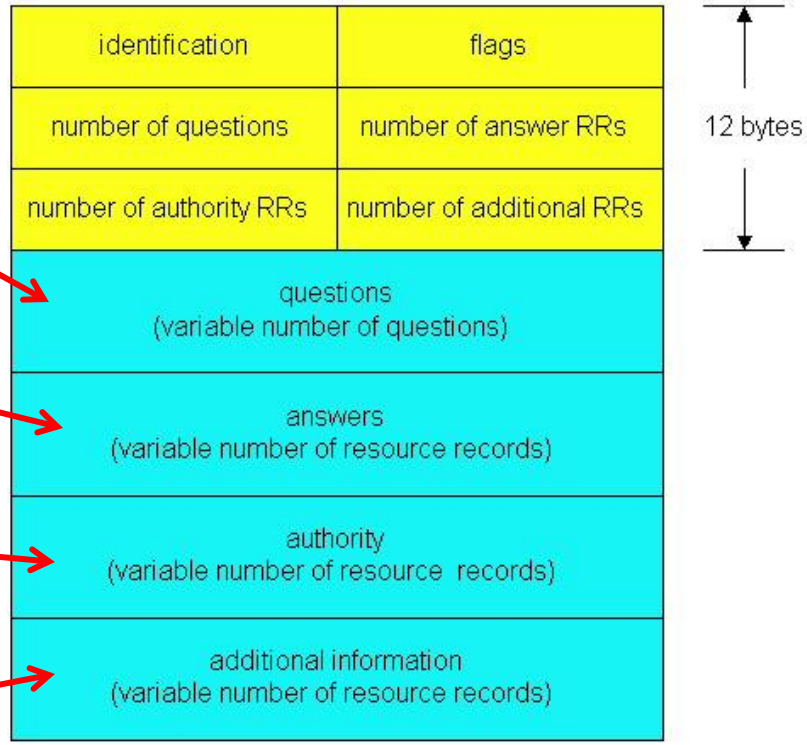
DNS protokol

meno a typ v otázke

RR záznamy v odpovedi

záznamy autoritatívnych serverov

dodatočné záznamy



DNS server

- počúva na porte 53
- komunikácia cez TCP alebo UDP
  - tí, čo žiadajú o preklad, obvykle komunikujú cez UDP
  - zrkadlá žiadajúce o všetky autoritatívne informácie komunikujú cez TCP
- ak chcem nový autoritatívny DNS server pre nejakú doménu, napríklad pre **psin.sk**, musí prevádzkovateľ TLD DNS servera pre doménu sk vložiť nasledovné údaje:

```
(psin.sk, ns.psin.sk, NS)
(ns.psin.sk, 10.10.10.10, A)
```

- Na serveri ns.psin.sk si môžem vytvárať záznamy typu A, napríklad www.psin.sk, alebo aj MX záznamy, ak chcem prevádzkovať aj mailový server

Zdieľanie súborov cez P2P

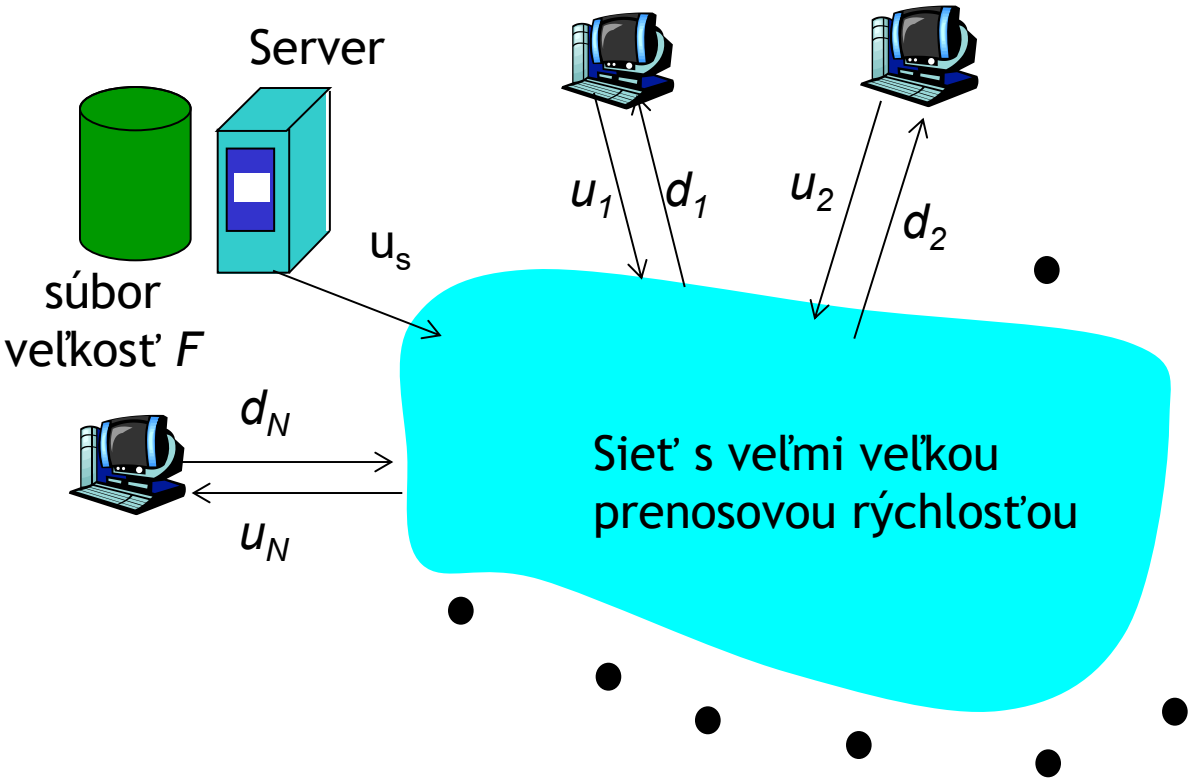
Príklad

- Alica spúšťa P2P program na svojom notebooku
- bežne sa pripája na rôznych miestach a stále dostane inú IP adresu
- hľadá film “Matrix”
- aplikácia jej zobrazí, ktorý z peerov v jej P2P sieti zdieľa film Matrix
- Alica si vyberie spomedzi peerov Boba
- súbor sa kopíruje z Bobovho počítača do Alicinho notebooku
- zatiaľ čo Alica sťahuje film Matrix, iní peerovia môžu ťahať zdieľané dáta z jej notebooku
- Alicin P2P program je zároveň klientom aj serverom.

Všetci peerovia sú servery = vysoká škálovanosť!

Porovnanie klient/server a P2P architektúry

Otázka : Koľko času je potrebné na distribúciu súboru pôvodne z jedného servera na *N* počítačov?



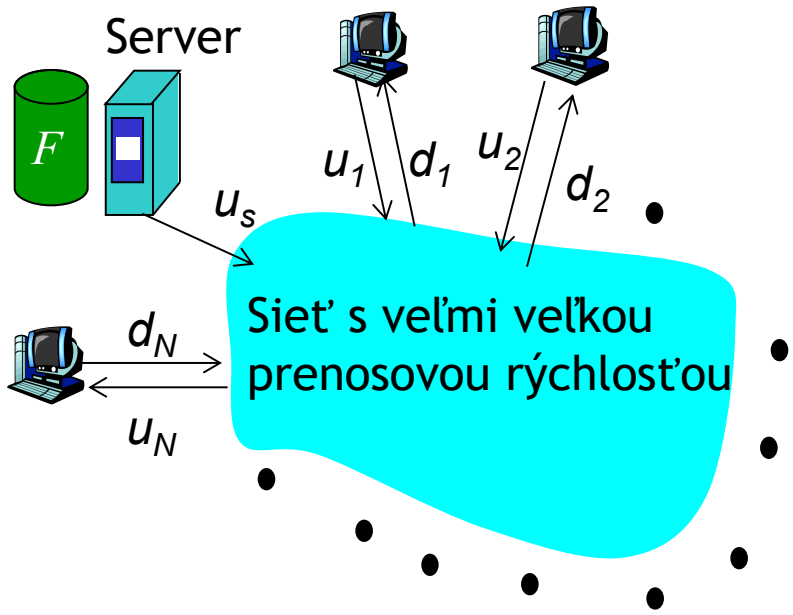
u_s: šírka pásma servera pre upload

u_i: šírka pásma klienta alebo peera pre upload

d_i: šírka pásma klienta alebo peera pre download

Klient/server: čas distribúcie súboru

- server odosielajúci *N* kópií súboru veľkosti *F* sekvenčne:
  - NF/u_s sekúnd
- Klient *i* potrebuje minimálne F/d_i sekúnd na download



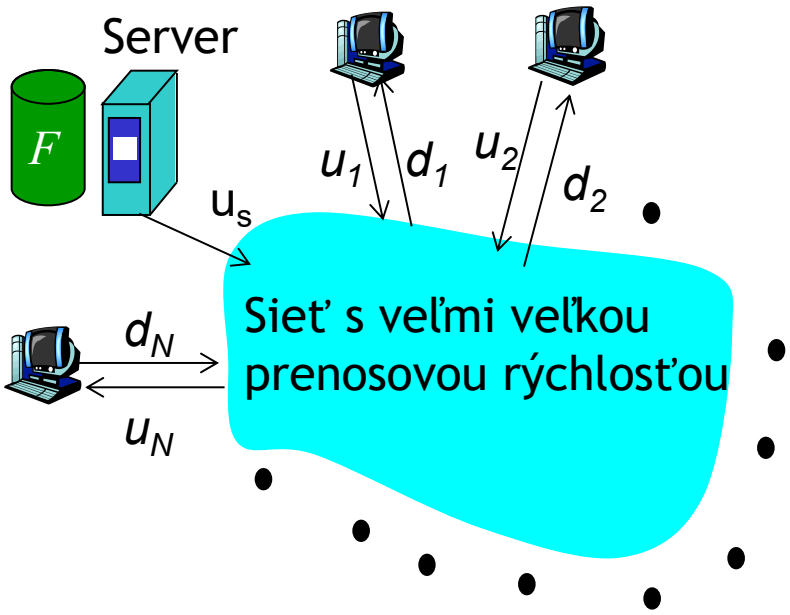
Čas na rozposlanie súboru veľkosti *F* ku *N* klientom

$$= d_{ks} = \max \{ NF/u_s, F/\min_i(d_i) \}$$

pre veľké N rastie lineárne

P2P: čas distribúcie súboru

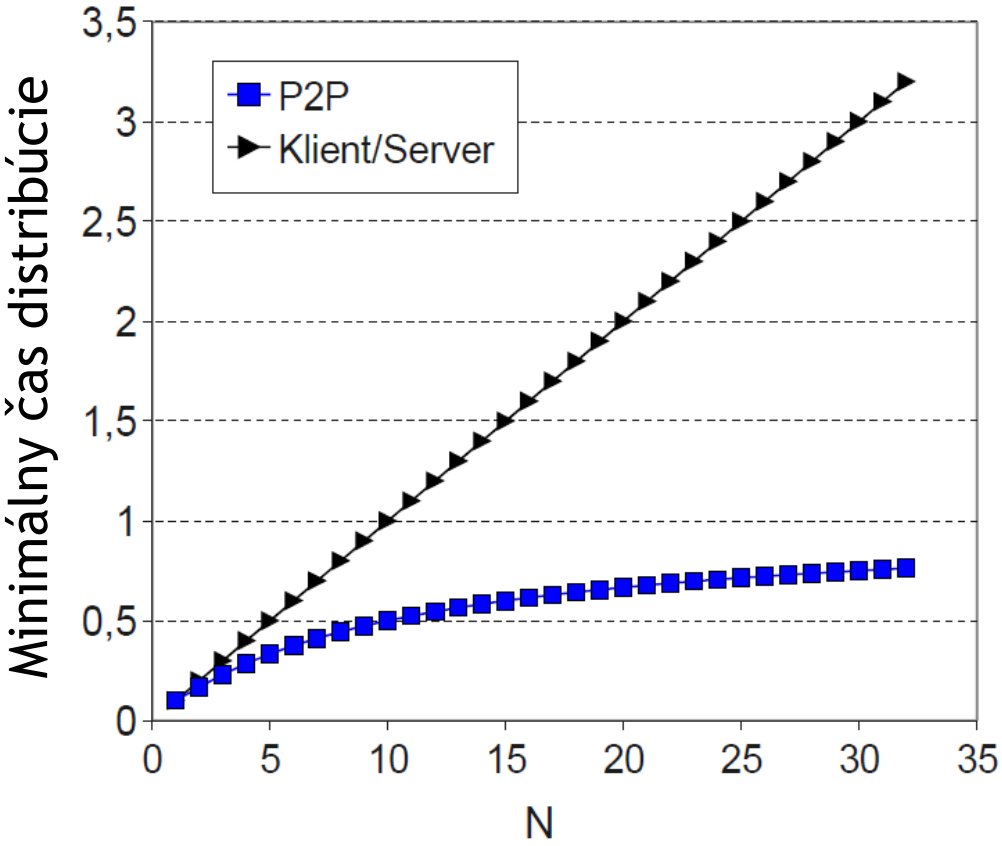
- server musí poslať minimálne jednu kópiu: F/u_s sekúnd
- Klient *i* potrebuje minimálne F/d_i sekúnd na download
- NF bitov musí byť stiahnutých na peerov dokopy



- Najrýchlejšie odosielanie zo všetkých peerov súčasne tvorí šírku pásma $u_s + \Sigma u_i$

$$d_{P2P} = \max \{ F/u_s, F/\min_i(d_i) , NF/(u_s + \sum_i u_i) \}$$

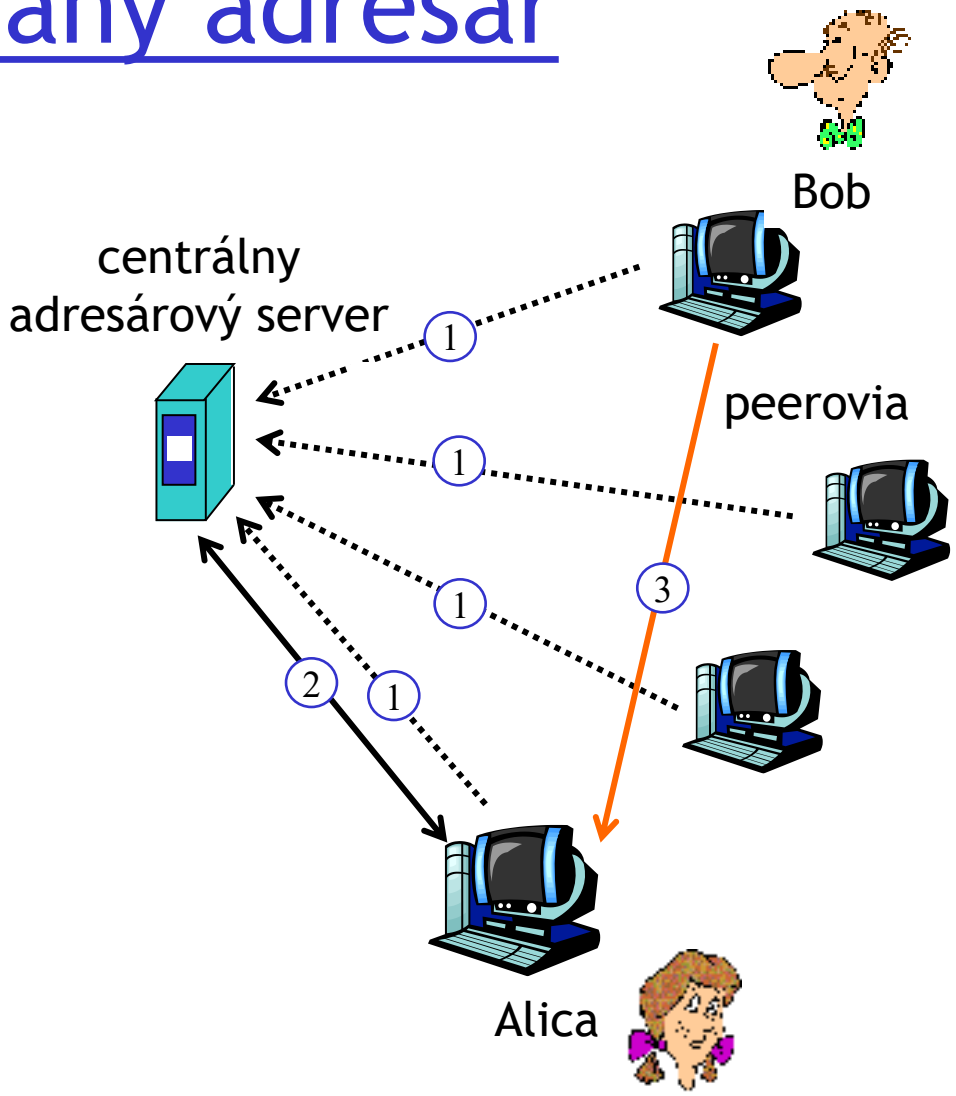
Porovnanie klient/server a P2P architektúry



P2P: centralizovaný adresár

dizajn pre “Napster”

1) peerovia sa napájajú kontaktovaním na centrálny server:
   - známa IP adresa
   - index obsahu zdieľaných adresárov
2) Alica na serveri vyhľadá “Matrix”
3) Alica osloví Boba a požiada ho o zaslanie súboru “Matrix”



P2P: problémy s centrálnym adresárom

- single point of failure
- slabé miesto výkonu
- porušovanie copyrightov: prevádzkovateľ je jasným “cieľom” pre právne kroky

Prenos súborov je decentralizovaný, ale vyhľadávanie je centralizované

Query flooding: Gnutella

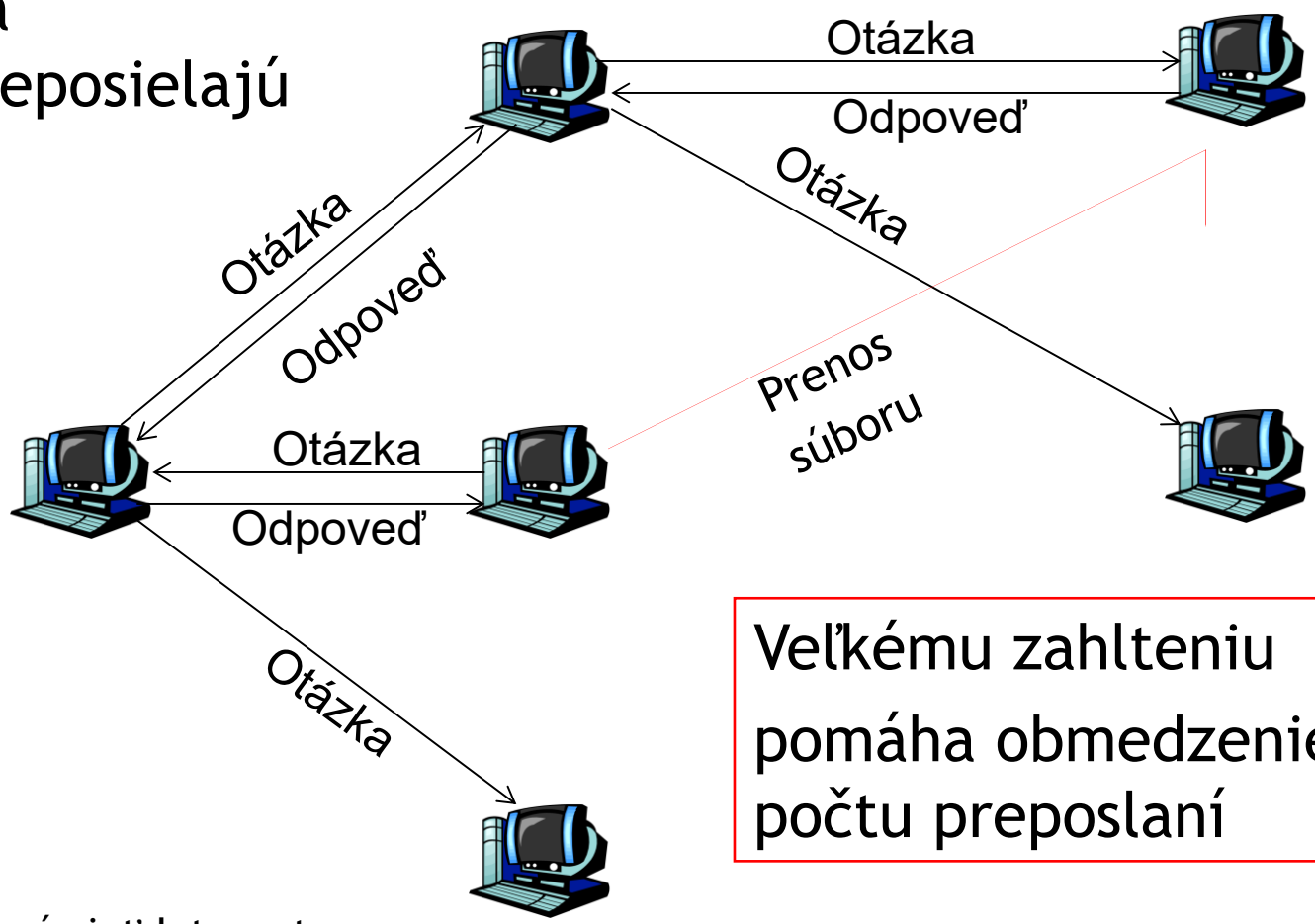
- "zaplavovanie požiadavkou"
- plne distribuovaná architektúra
  - bez centrálneho servera
- verejný protokol
- Gnutella má veľa implementácií klientov

graf spojení klientov

- hrana v grafe = existujúce TCP spojenie medzi klientami X a Y
- hrana je virtuálne spojenie uzlov, nie fyzické
- jeden peer je typicky spojený hranou s menej ako 10 susedmi

Gnutella: protokol

- otázka je zaslaná cez existujúce TCP spojenia
- peerovia preposielajú otázky ďalej
- odpoveď je zasielaná rovnakou cestou späť



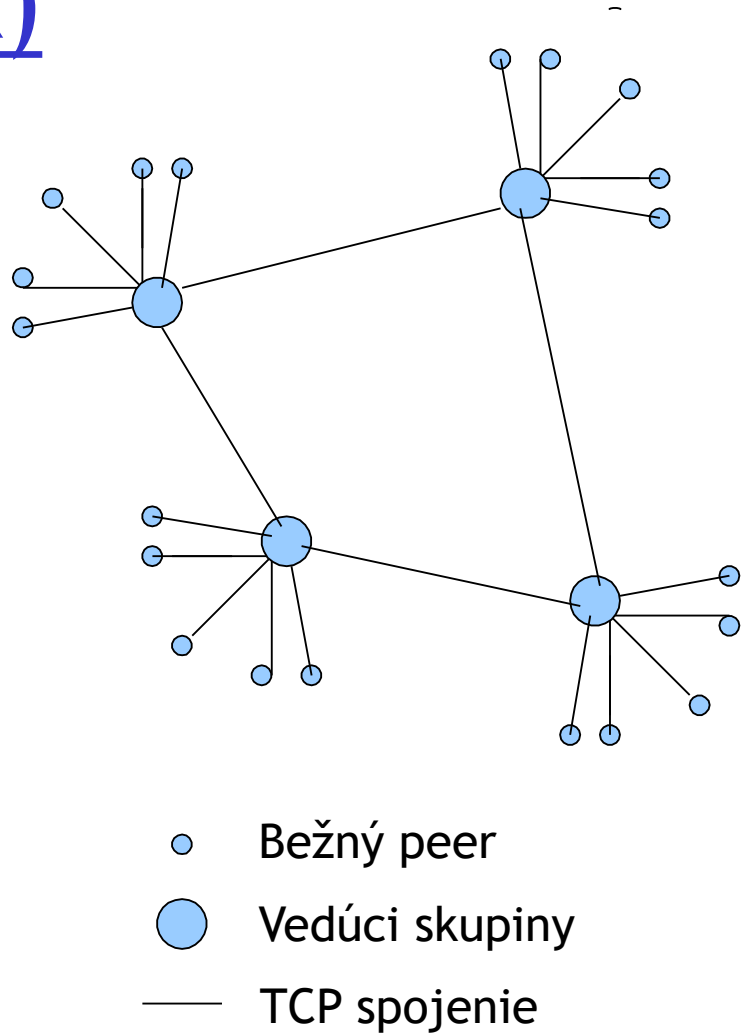
Veľkému zahlteniu pomáha obmedzenie počtu preposlaní

Gnutella: Pripájanie peerov

- Keď sa Alica chce pripojiť, musí sa napojiť na niektorého už napojeného peera v sieti Gnutella: použije zoznam kandidátov
- Alica skúša kandidátov v zozname, až sa jej podarí vytvoriť TCP spojenie napr. s Bobom
- Zaplavenie: Alica pošle Bobovi správu Ping
- Bob preposiela správu Ping svojim susedom v sieti, ktorí ju preposielajú ďalej, atď.
- "Niektorí" peerovia, ktorí príjmu správu Ping, odpovedia Alici správou Pong
- Alica v svojej Ping správe prikladá vzdialenosť (počet preposlaní), z ktorej chce prijať Pong správy
- Alica prijme správy Pong a môže vytvoriť s niektorými nové TCP spojenia

Hierarchická sieť (Kazaa a jej protokol FastTrack)

- kompromis medzi centralizovaným indexom a zaplavovaním požiadavkami
- každý peer je buď vedúci skupiny, alebo pripojený na vedúceho skupiny
  - TCP spojenia medzi peermi a ich vedúcim skupiny
  - TCP spojenia medzi niektorými vedúcimi skupiny
- Vedúci skupiny indexuje obsah adresárov svojich členov
- Vedúci skupín medzi sebou používajú zaplavovanie

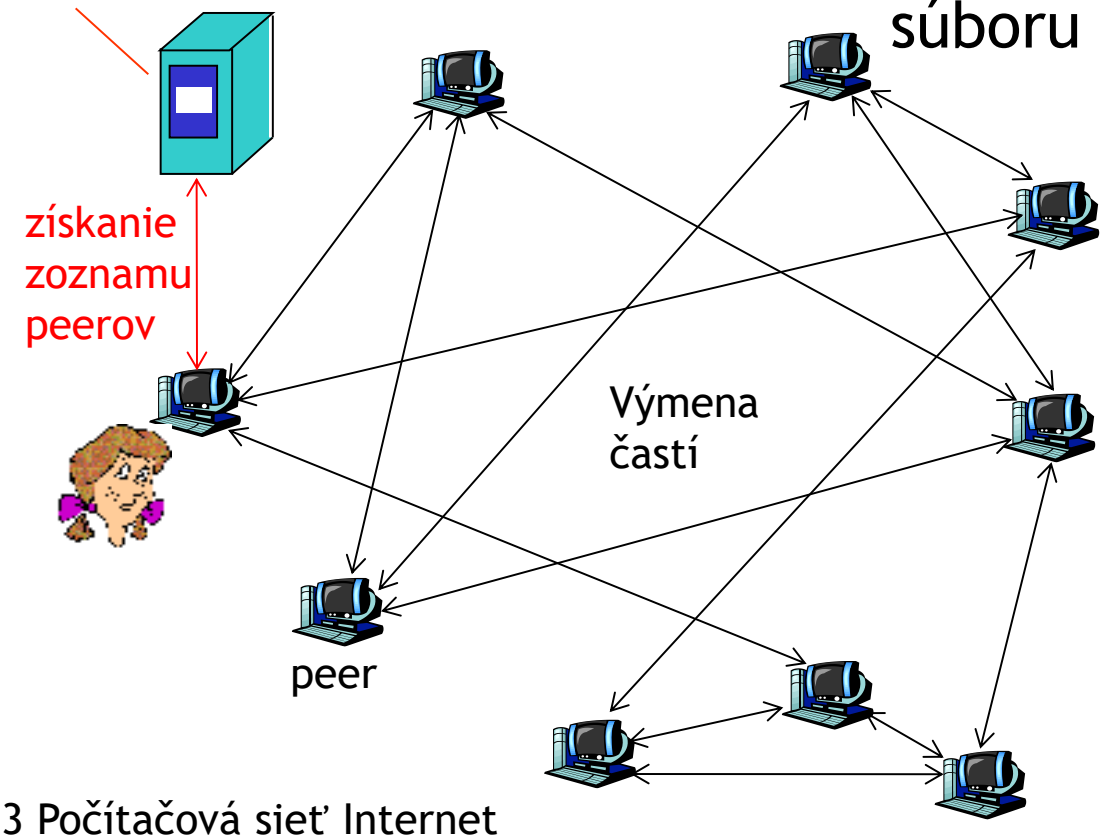


BitTorrent

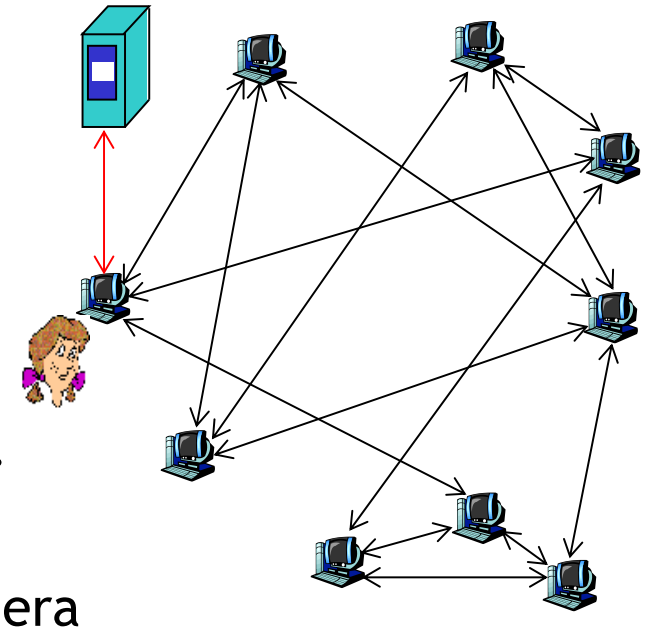
- distribúcia súborov cez P2P

tracker: registruje peerov zúčastnených na distribúcii

torrent: skupina peerov vymieňajúca si časti súboru



BitTorrent



- súbor je rozdelený na malé časti
- = *chunks* *(mocnina 2, typicky 256 KB – 4 MB).*
- peer sa napojí na torrent:
  - nemá žiadne časti, ale časom si nejaké zozbiera
  - je registrovaný u trackera na získanie zoznamu peerov, na ktorých sa napojí
  - alternatívne cez peer exchange/local peer discovery
- počas sťahovania, peer posiela na požiadanie tie časti, ktoré už má, iným peerom
- peerovia sa môžu pripájať a odpájať
- keď peer doťahal celý súbor, môže sa (sebecky) odpojiť alebo (obetavo) ostať napojený = seeder

BitTorrent

Výber častí na sťahovanie:

- V ľubovoľnom čase majú peerovia rôzne podmnožiny častí
- Peer (Alica) stále dookola žiada svojich susedov o zoznam častí, ktoré majú
- Alica generuje žiadosti o tie časti, ktoré zatiaľ nemá
  - najskôr najzriedkavejšie

Odosielanie častí (príklad):

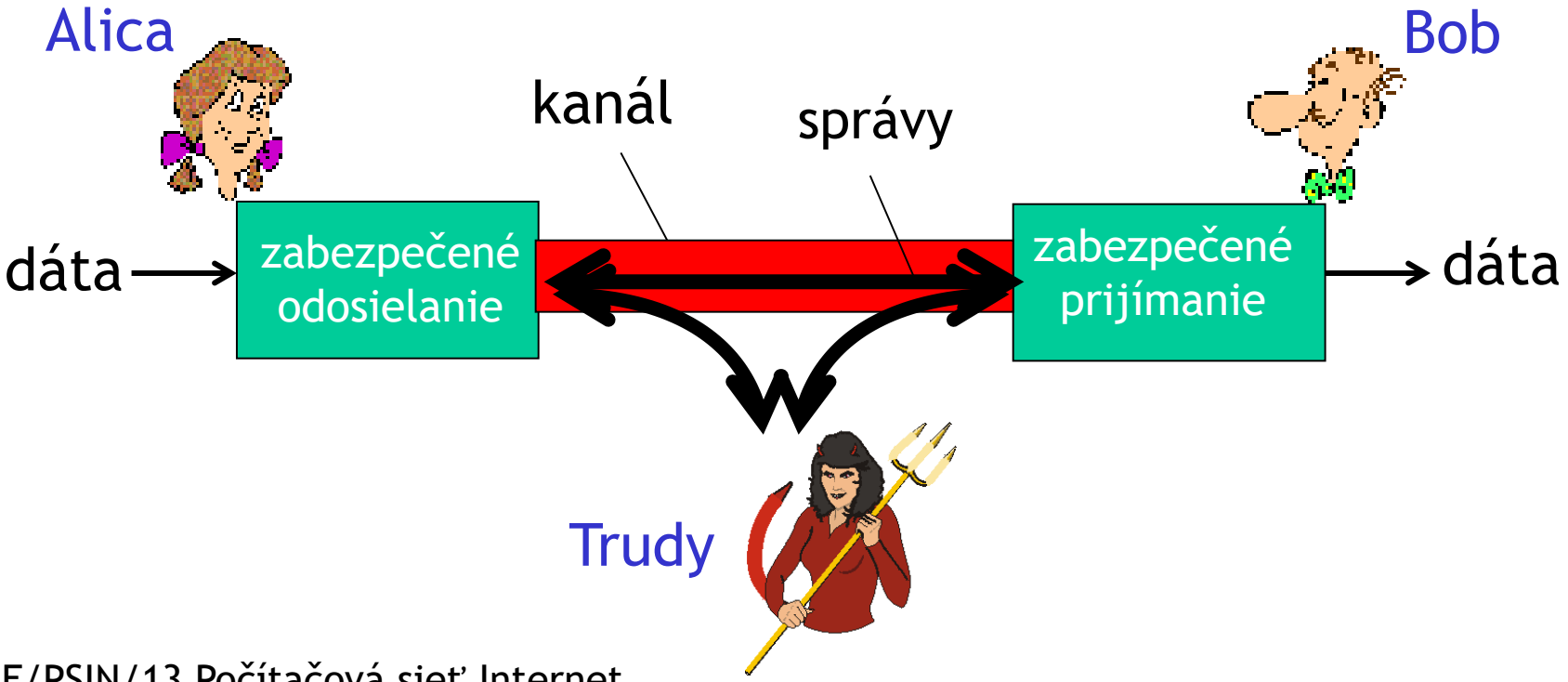
- Alica odosiela časti štyrom žiadajúcim susedom, ktorí jej posielajú svoje časti *najvyššou rýchlosťou*
  - Prepočítava najlepších 4 každých 10 sekúnd
- každých 30 sekúnd si vyberie jedného náhodného žiadateľa a pošle mu, čo žiadal
  - Tento peer sa môže dostať medzi najlepších štyroch

Zabezpečenie protokolov

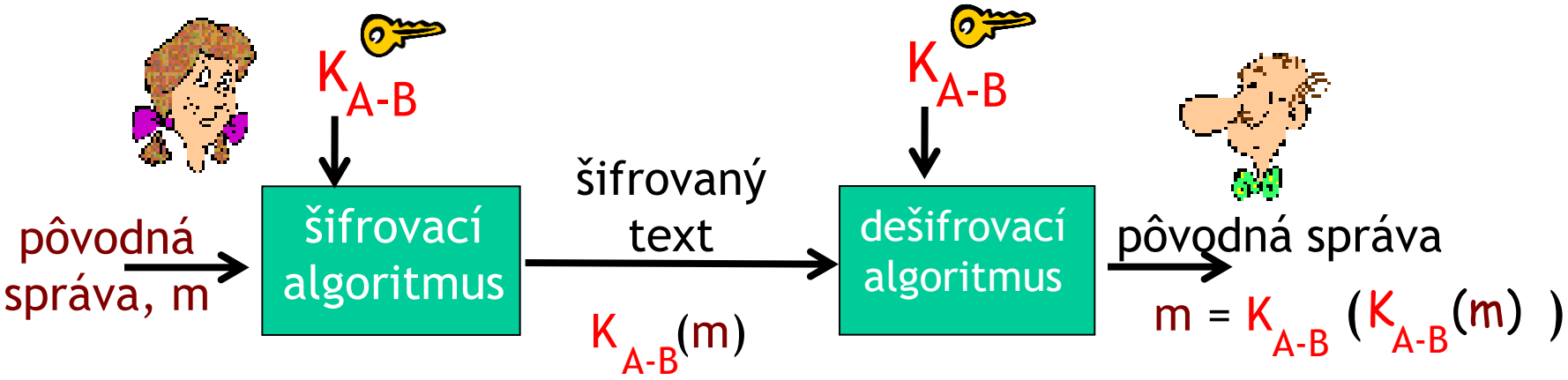
- zadefinujeme dôležité pojmy
  - symetrické šifrovanie
  - asymetrické šifrovanie, súkromný a verejný kľúč
  - elektronický podpis
  - poskytovateľ dôveryhodných služieb (certifikačná autorita)
- príklad zabezpečenia pre aplikačné protokoly
- ...iba “slabý vývar” bezpečnosti na internete

Priatelia a útočníci: Alica, Bob, Trudy

- známe mená vo svete bezpečnosti sietí
- Alica a Bob chcú "bezpečne" komunikovať
- Trudy (útočník) môže odpočúvať, meniť, mazať a pridávať správy do ich spojenia

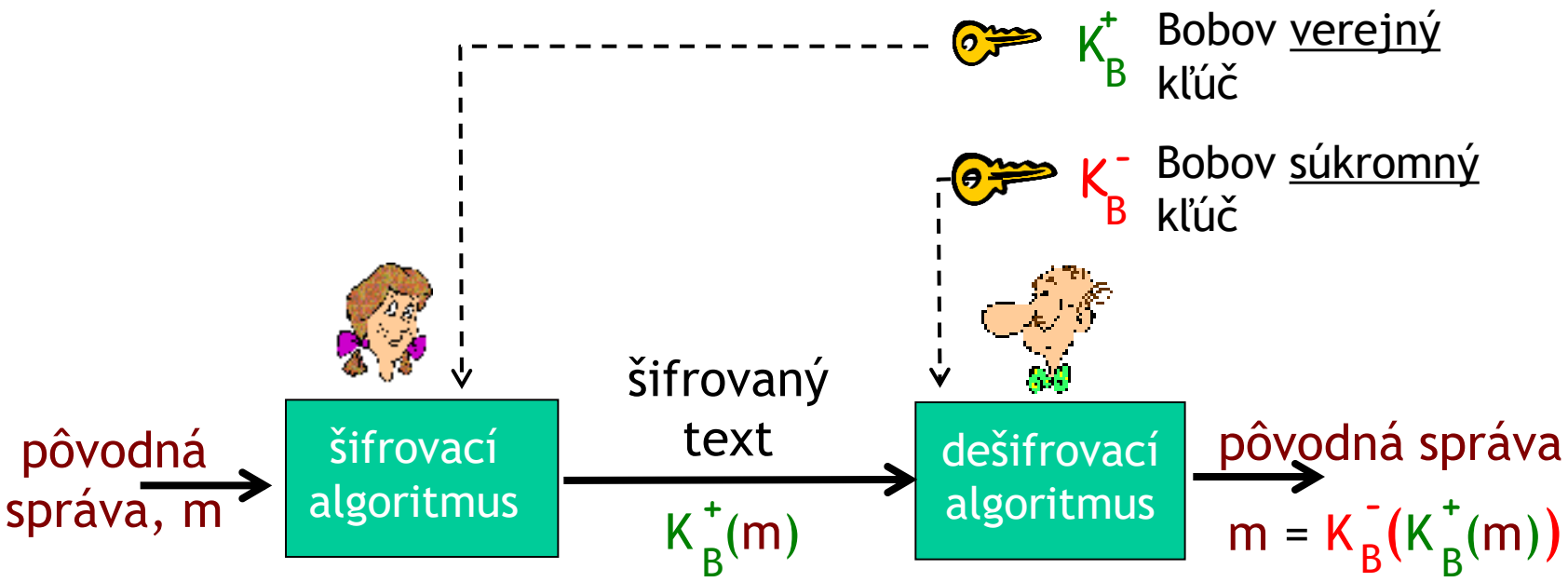


Symetrická kryptografia



- Symetrická kryptografia: Alica a Bob zdieľajú rovnaký symetrický kľúč: K_{A-B}
- Otázka: Ako si Alica a Bob dohodnú spoločný kľúč, ak ho zatiaľ dohodnutý nemajú?

Asymetrická kryptografia



Vlastnosti súkromného z verejného kľúča

Požiadavky:

1. Potrebujeme $K_B^+( )$ a $K_B^-( )$ také, že

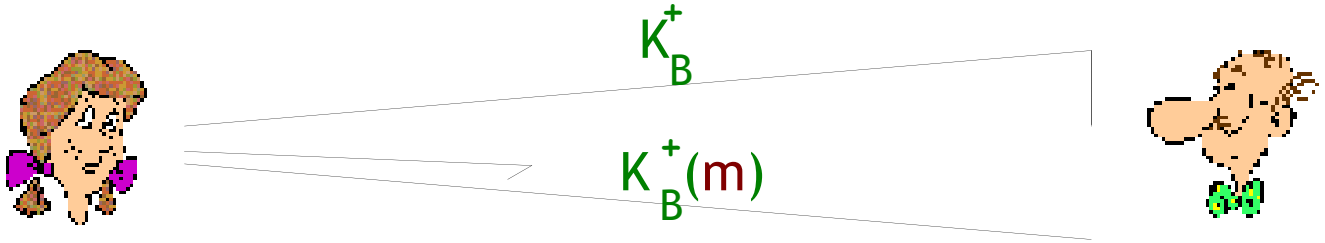
$$K_B^-(K_B^+(m)) = m$$

2. Ak máme verejný kľúč K_B^+, malo by byť nemožné vypočítať z neho súkromný kľúč K_B^-

Napr. pre algoritmus RSA: Rivest, Shamir, Adleman platí aj užitočný opačný vzťah: $K_B^+(K_B^-(m)) = m$

Problém distribúcie verejného kľúča

- Ako si má Alica bezpečne stiahnuť Bobov verejný kľúč?
- Keďže je verejný, Bob ho môže dať hocikomu - ak ho niekto iný prečíta, tak to nevadí
- Zavesí ho na web?
- Pošle ho ako prvú správu spoločnej komunikácie?
- Ale čo ak ho niekto medzi nimi vymení?



$$K_B^-(K_B^+(m)) = m$$

Man-in-the-middle

K_B^+

$K_B^+(m)$

$K_B^-(K_B^+(m)) = m$

K_B^+

K_T^+

$K_T^+(m)$

$K_T^-(K_T^+(m)) = m$

$K_B^+(m)$

$K_B^-(K_B^+(m)) = m$

Šifrovacia hash funkcia

- vezmeme ľubovoľne dlhú správu m, vygenerujeme hodnotu H(m) fixnej dĺžky, ktorú následne môžem zašifrovať s použitím kľúča K
- vlastnosti šifrovacej hash funkcie:
  - malo by byť výpočtovo ťažké nájsť dve rôzne správy x a y také, že H(x) = H(y)
  - na základe znalosti H(m) by sme nemali vedieť získať m

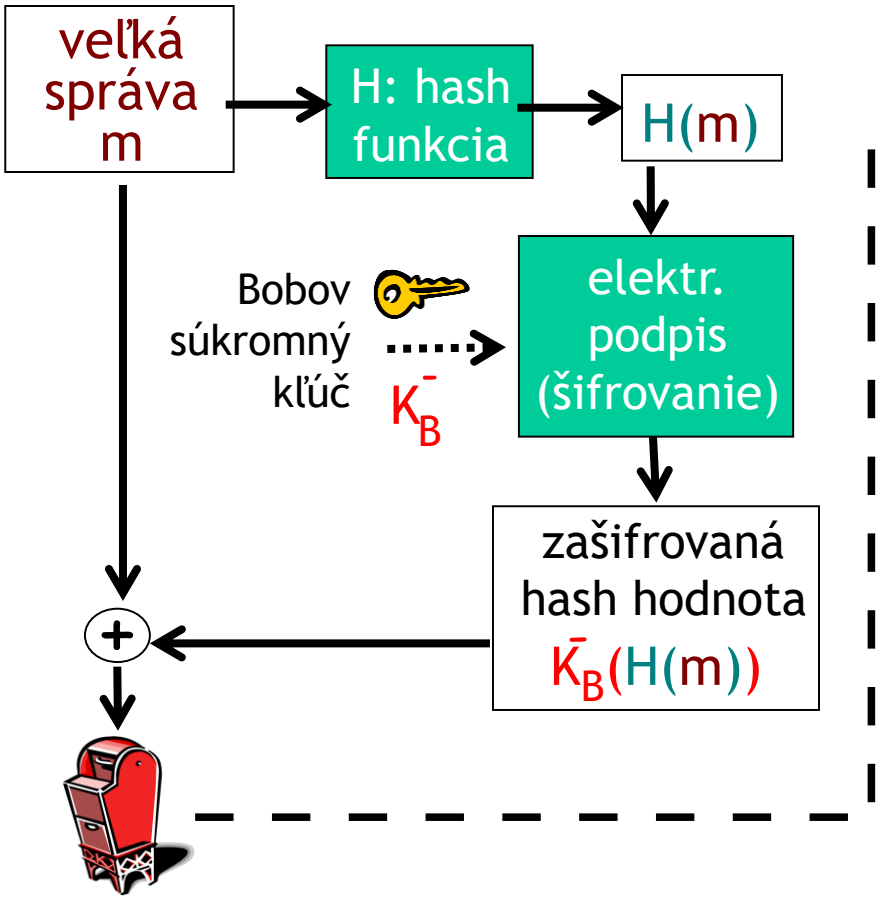
Elektronický podpis

šifrovacia technika analogická s ručným podpisom

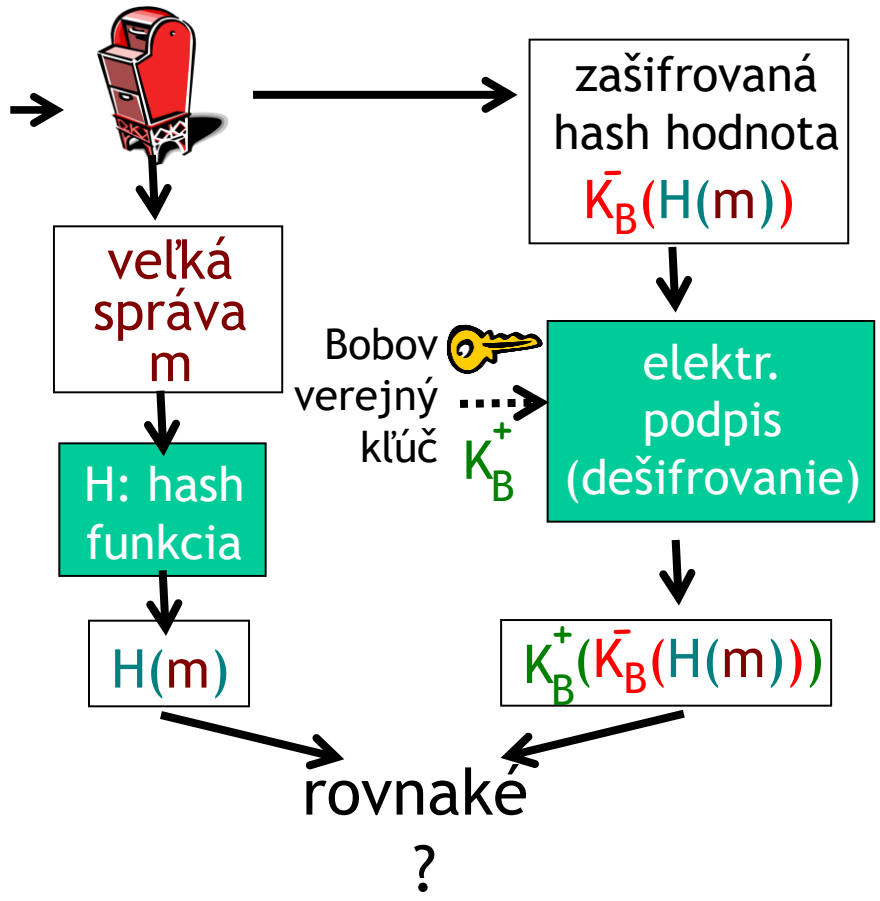
- odosielateľ (Bob) elektronicky podpíše jeho dokument deklarujúc, napr. že je autorom dokumentu
- podpis sa prikladá k pôvodnému (nezašifrovanému) dokumentu
- verifikovateľnosť: vieme dokázať, že dokument je podpísaný Bobovým podpisom
- nepopierateľnosť: vieme tvrdiť, že podpis tohto dokumentu nemohol byť vygenerovaný niekým iným ako Bobom

Elektronický podpis

Bob odošle elektronicky podpísanú správu:



Alica overí pravosť podpisu a neporušenie elektronicky podpísanej správy:



Certifikovanie verejného kľúča

problém s verejným kľúčom:

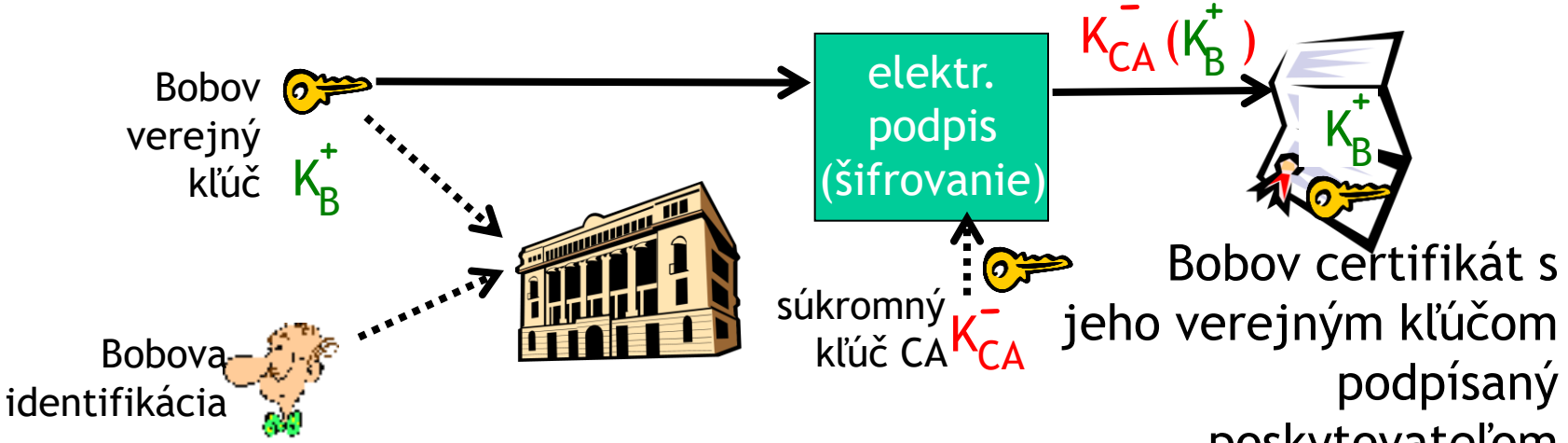
- Keď Alica získa Bobov verejný kľúč (z webstránky, e-mailu), ako môže *vedieť*, že je to Bobov a nie Trudyn verejný kľúč?

riešenie:

- poskytovateľ dôveryhodných služieb (certifikačná autorita)

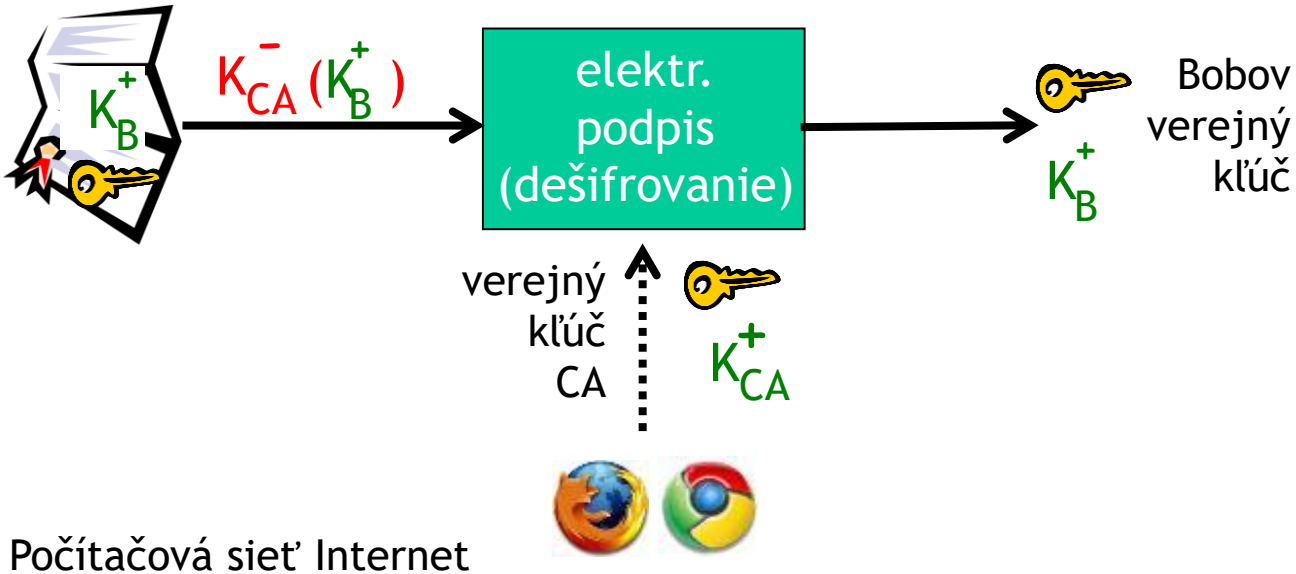
Poskytovateľ dôveryhodných služieb

- Poskytovateľ dôveryhodných služieb (CA): spája entitu E (počítač, osobu) s jej verejným kľúčom.
- E si registruje svoj verejný kľúč u CA.
  - E poskytne "dôkaz identity" pre CA.
  - CA vytvorí certifikát spájajúci E a jeho verejný kľúč
  - certifikát je elektronicky podpísaný poskytovateľom dôveryhodných služieb, ktorý tvrdí: "Toto je verejný kľúč patriaci E."



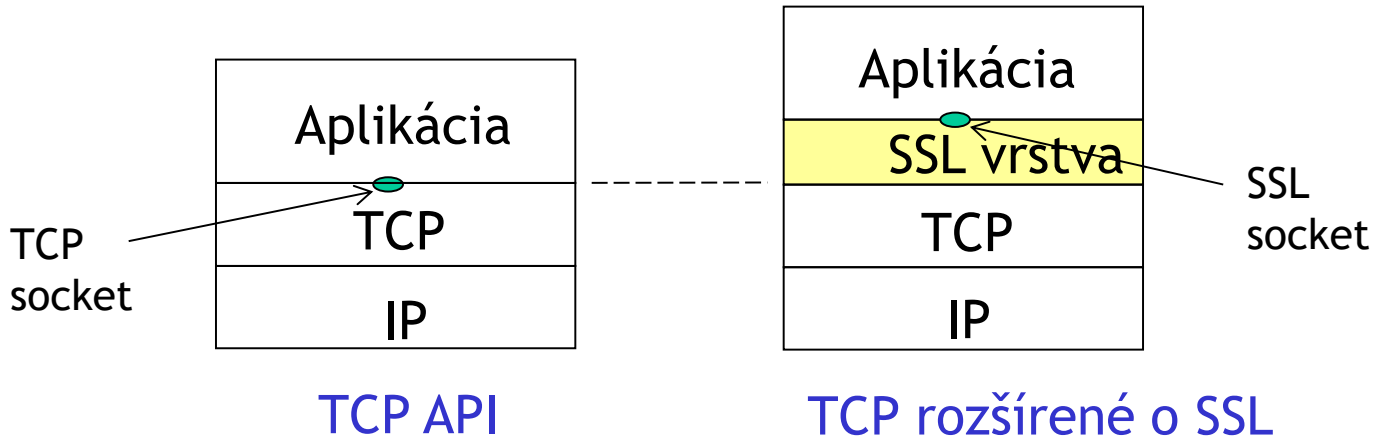
Poskytovateľ dôveryhodných služieb

- Keď Alica chce získať Bobov verejný kľúč:
  - vezme si Bobov certifikát (od Boba alebo inak).
  - použije verejný kľúč CA a získa tak overený Bobov verejný kľúč
  - verejné kľúče kvalifikovaných poskytovateľov dôveryhodných služieb bývajú súčasťou inštalácie sieťových programov (napr. web. prehliadačov)



Secure sockets layer (SSL)

- poskytuje zabezpečený kanál pre ľubovoľnú sieťovú aplikáciu založenú na TCP protokole
  - Napr. HTTPS, FTPS, VoIP, IMAPS, POP3S, XMPP
- jeho nástupca je TLS: Transport Layer Security aktuálne vo verzii 1.3 (RFC 8446)
- radí sa do transportnej vrstvy
- poskytuje služby:
  - Autentifikácia servera, šifrovanie dát, autentifikácia klienta (voliteľná)

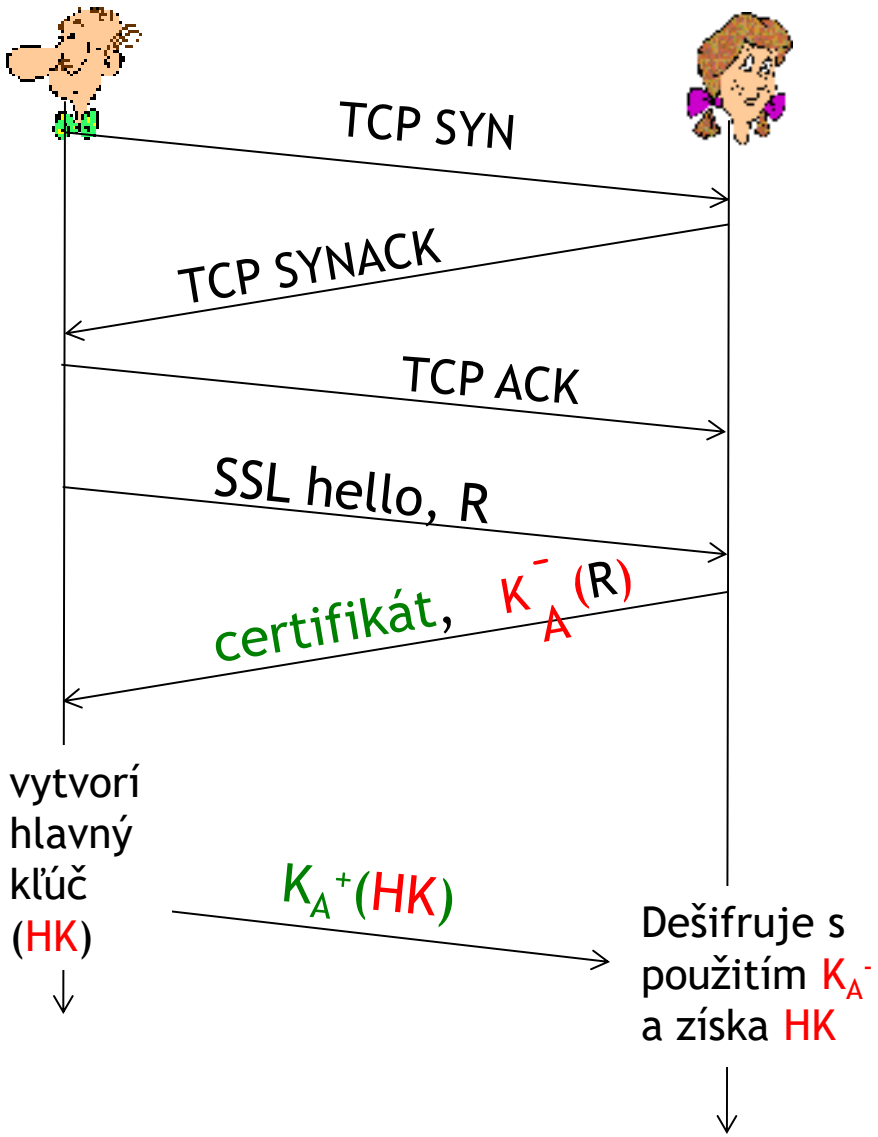


Vlastnosti SSL protokolu

- Súkromné spojenie
  - Všetky správy sú šifrované symetrickým šifrovacím kľúčom vytvoreným nanovo pre každé nové spojenie
- Zabezpečené a spoľahlivé dohadovanie šifrovacieho kľúča
  - šifrovací kľúč pre komunikáciu môže byť známy iba komunikujúcim stranám
  - žiaden útočník nemôže do tohto dohadovania zasiahnuť bez toho, aby bol odhalený
- Integrita správ
  - Každá správa je podpísaná pomocou šifrovacej hash funkcie
- Autentifikácia komunikujúcich
  - S použitím certifikátov

SSL: idea dohadovania hlavného kľúča

- Bob iniciuje TCP spojenie s Alicou a pošle jej náhodnú hodnotu R
- autentifikuje Alicu cez jej certifikát podpísaný certifikačnou autoritou a dešifrovaním získa R s pomocou jej verejného kľúča
- vytvorí symetrický kľúč HK a zašifruje ho Alicinim verejným kľúčom - výsledok pošle Alici



Ďakujem za pozornosť

Modifikované slajdy z knihy:

Computer Networking: A Top Down Approach ,
4th edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2007.