

Operační systémy a sítě

Petr Štěpán, K13133

KN-E-229

stepan@labe.felk.cvut.cz

Téma 12.

DNS, VPN, NAT a firewally

Domain Name System (DNS)

- Uživatelé preferují symbolická jména strojů
 - oproti číselným adresám
 - Nutno zajistit mapování **jméno_stroje** \Leftrightarrow **číselná_adresa**
- Internet používá
 - hierarchický prostor jmen (*namespace*)
 - nezávislý na topologii sítí
 - Citát: *V TCP/IP Internetu jsou hierarchicky členěná jména strojů přiřazována podle struktury organizací, které mají oprávnění nakládat s částmi prostoru jmen, nikoli však nutně podle způsobu propojení fyzických sítí.*
- Flexibilní hierarchie jmen
 - symbolická jména strojů – **doménová jména** (*domain name*)
 - celosvětově jednotná syntaxe
 - hierarchie vyznačena oddělovačem '.'
 - např. `cyber.felk.cvut.cz`
 - vše, co následuje za první '.' se obvykle nazývá doména
 - poslední úsek ('cz') je tzv. doména 1. řádu nebo vrcholová doména (*top-level domain* = TLD)

Základní TLD

- Původní historicky vzniklé v USA a Kanadě
 - stále užívané

com Komerční organizace (commmercial)

edu Vzdělávací instituce (educatinal)

gov Vládní jednotky (government)

mil Vojenské užití (military)

int Mezinárodní organizace (international)

org Původně neziskové organizace

net Organizace sítě (network) – v současnosti i širší užití

arpa Historická doména, nyní používaná pro tzv. „reverzní rezoluci“ →

národní kód

Geografické členění podle států – dvoupísmenné kódy				
	cz	Česko	uk	Velká Británie
	de	Německo	at	Rakousko
	us	USA (málo užívané)	cn	Kontinentální Čína

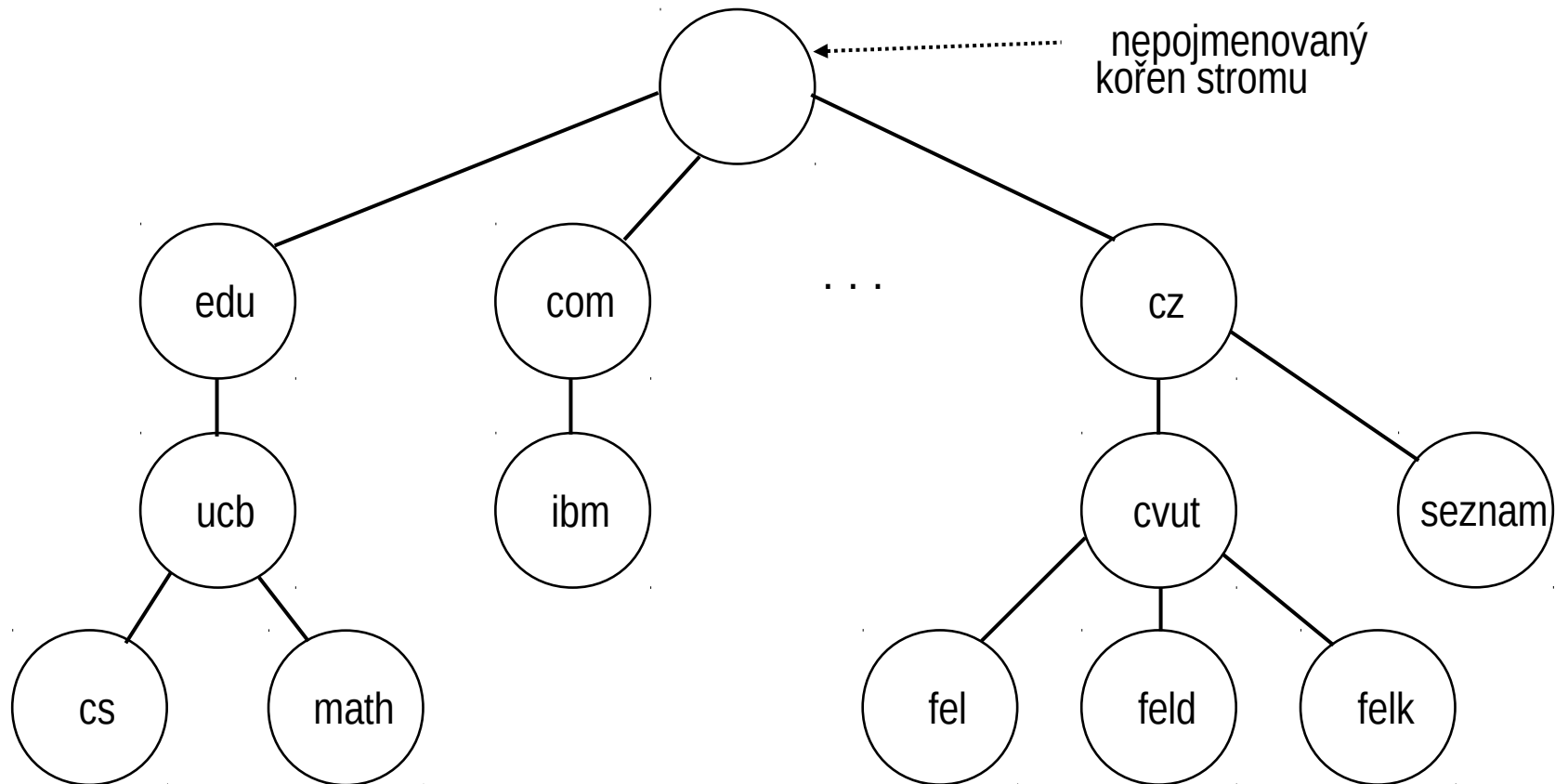
- Přidělování domén 2. řádu je svěřeno národním autoritám
 - V ČR je to CZ-NIC



– V některých zemích jsou domény 2. řádu účelově orientované

• Např. co.uk (britské komerční domény), ac.uk (akademické instituce)

Hierarchie DNS



- **Autorizace a správa jmen ve směru hierarchie**

- Např. cvut se registruje u správce TLD cz a spravuje cvut.cz
- Správci sítě FEL na Karlově náměstí registrují felk u správců cvut.cz a jsou autorizováni přidělovat jména v doméně felk.cvut.cz

Databáze domén a doménových jmen

- Celosvětová hierarchicky organizovaná distribuovaná databáze
 - Záznamy jsou trojice (jméno, třída, obsah)
 - Třída označuje typ objektu, který záznam popisuje
 - Např. "stroj", "poštovní server" (*mail exchanger*) atd.
 - Důsledek: *Jedno jméno může označovat více různých objektů. Klient, který chce získat informaci z databáze, uvádí i typ požadovaného objektu.*
- Terminologie:
 - Servery DNS se označují jako **jmenné servery** (*name servers*)
 - DNS software na klientských strojích se nazývá **rezolver** (*resolver*)
- DNS používá množinu on-line serverů ve stromové struktuře
 - Daný server může obhospodařovat celý podstrom nebo jen jednu či několik vrstev
 - Např. kořenový server má informace o všech TLD a jejich jmenných serverech

Základní typy DNS záznamů

Typ	Pojmenování	Obsah záznamu
SOA	Start of Authority	Sada polí určujících kterou část merné hierarchie "teno" server implementuje
A	Host Address	32-bitová IP adresa
CNAME	Canonical Name	Kanonicke doménové jméno pro "přezdívku" (alias)
NS	Name Server	Jméno autoritativního serveru této domény
MX	Mail Exchanger	16-bitové preferenční číslo a jméno stroje, který pracuje jako hlavní mail server pro tuto doménu
PTR	Pointer	Doménové jméno pro tzv. inverzní rezoluci
AAAA	Host Address	128-bitová IPv6 adresa

- Ukázka jednoduché konfigurace jmeného serveru

```

$ORIGIN firma.cz.
$TTL      12h
@          SOA ns.domena.cz. spravce.mail.domena.cz. (
          2011090801 ; Serial
          3h         ; Slave refresh (3 hours)
          1h         ; Slave retry time in case of a problem (1 h)
          2h         ; Slave expiration time (2 days)
          3600       ; Maximum caching time of failed lookups (1 h)
          )
firma.cz.  NS      ns.firma.cz.
firma.cz.  NS      ns2.iol.cz.
firma.cz.  MX      0 mail.firma.cz.
firma.cz.  MX      100 relay.iol.cz.

ns         A      190.18.113.16
mail       A      190.18.113.16
www        A      190.18.113.14
extern-site A      147.120.198.155
    
```

Řešení úlohy hledání doménových jmen

- Je nutno hledat doménová jména od kořene stromu?
- V praxi
 - Vyhledávání začíná u lokálního jmenného serveru
 - Každý stroj v lokální síti musí znát jeho adresu
 - Lokální server se bude obracet na kořenový server jen zcela výjimečně
 - Bude se obracet na svůj nadřazený server
 - Např. lokální server na Karlově náměstí ns.felk.cvut.cz se bude obracet na ns.cvut.cz
 - Rekurzivní postup "směrem nahoru"
- **Problém efektivity**
 - Fakta:
 - nejčastější dotazy jsou lokální
 - pár [jméno – adresa] se mění zřídka
 - dotazy se často opakují
 - Řešení:
 - Každý server si pamatuje (v cache) odpovědi, které získal od nadřazených serverů
 - Dobu platnosti pamatovaného údaje (TTL) udává zdrojový server, který opravdu drží původní datový záznam (tzv. **autoritativní server**)

Zkracování doménových jmen při hledání

- DNS pracuje s úplnými doménovými jmény
 - Např. cyber.felk.cvut.cz
- Rezolver však umožní hledaná jména zkracovat
 - např. na lokálním stroji je nastaveno přednostní vyhledávání v doménách
 - .felk.cvut.cz, .fel.cvut.cz, .cvut.cz
 - Zadá-li uživatel (nebo jeho aplikace) jméno pouze jako www, zajistí rezolver posloupnost dotazů (v tomto pořadí)
 - www.felk.cvut.cz, www.fel.cvut.cz, www.cvut.cz
 - a první pozitivní odpověď považuje za správnou (www.fel.cvut.cz)
- DNS umí mapovat jen úplná doménová jména na adresy
 - Zkratky jsou k dispozici pouze díky lokálním rezolverům z důvodů většího uživatelského pohodlí.

```
C:\> ipconfig /all
Windows IP Configuration
    Host Name . . . . . : zubrina
    Primary DNS Suffix . . : felk.cvut.cz
    DNS Suffix Search List. : felk.cvut.cz
                           fel.cvut.cz

Ethernet adapter Ethernet:
    Physical Address. . . . : 00-1A-A0-CF-6F-77
    IP Address. . . . . : 147.32.85.46
    Subnet Mask . . . . . : 255.255.255.128
    Default Gateway . . . . : 147.32.85.1
    DNS Servers . . . . . : 147.32.80.9, 147.32.1.20
```


Reverzní rezoluce

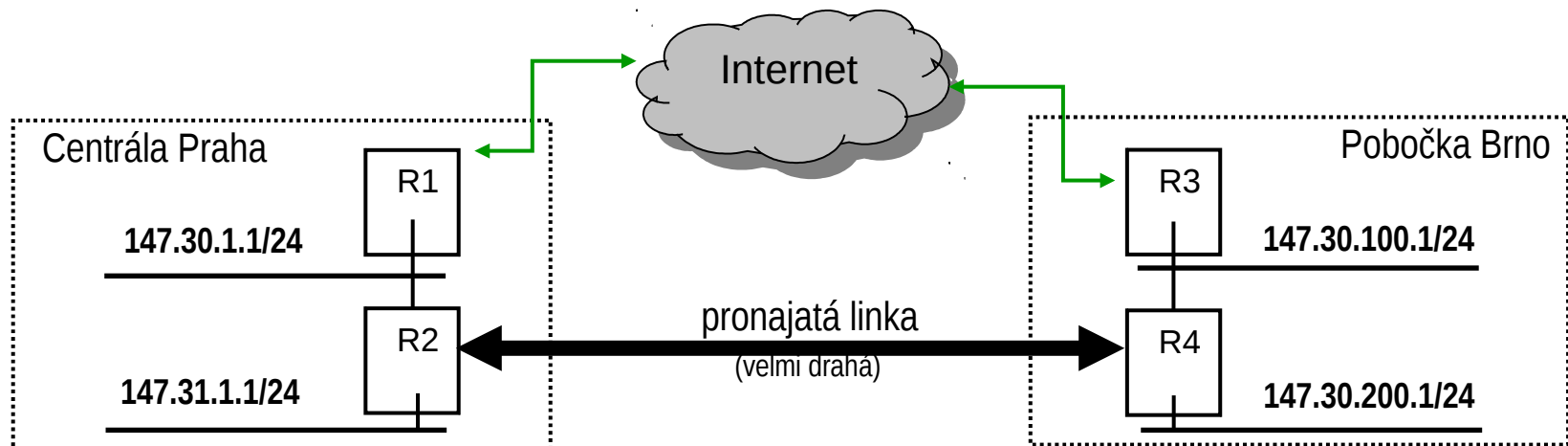
- Přímá rezoluce: doménové_jméno → IP_adresa
- Reverzní rezoluce: IP_adresa → doménové_jméno
 - Nemusí být jednoznačné
 - Více doménových jmen se může mapovat na jednu adresu
 - Poměrně neefektivní, avšak mnohdy potřebné
 - Např. anti-spam
 - Používají se PTR záznamy
- Trik realizovaný lokálním rezolverem
 - Zapiš adresu formálně jako jméno stroje a zeptej se na ně
 - Necht' IP adresa je *aaa.bbb.ccc.ddd*
 - Vytvoř dotaz na *ddd.ccc.bbb.aaa.in-addr.arpa*
 - Využití historické TLD *arpa* ←
 - Odešli name serveru, což je firewall, objekt typu PTR s

Reálný DNS systém

- Ukázali jsme jen princip DNS
- Skutečná implementace je výrazně složitější
 - požadavky na spolehlivost a efektivitu
- Realizace
 - Kořenových serverů je celá řada (spolehlivost, dostupnost)
 - Vzájemně si replikují informace o TLD serverech a "delegování" jejich autority
 - Primární a sekundární name-servery
 - Primární name-server je ten, který drží primární databázi informací o doméně
 - Sekundární server čas od času kopíruje obsah primárního serveru a zajišťuje autoritativní odpovědi, není-li primární server dostupný
 - "Caching-only" servery
 - Lokální doména (malá organizace) nemá svůj vlastní name-server
 - Ten je např. u poskytovatele připojení
 - Provozuje ale "caching-only" server, který nemá svoji vlastní databázi. Všechny dotazy přeposílá nadřazenému name-serveru, avšak jeho odpovědi si pamatuje a po dobu TTL dotazy vyřizuje lokálně
 - DDNS (Dynamické DNS)
 - Umožňuje mobilním strojům dynamicky registrovat svoje stabilní doménové jméno u poskytovatele připojení nebo u **dyndns.org**
 - Zatím málo rozšířené, zřídka implementováno na klientské straně

Virtuální privátní síť

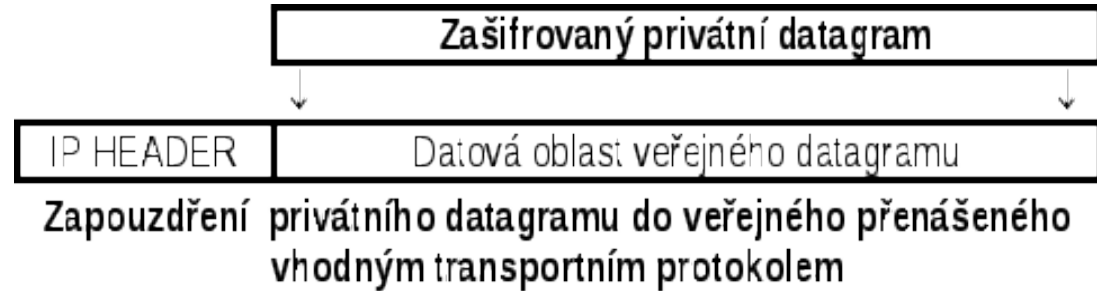
- Hybridní architektura organizace firma.cz



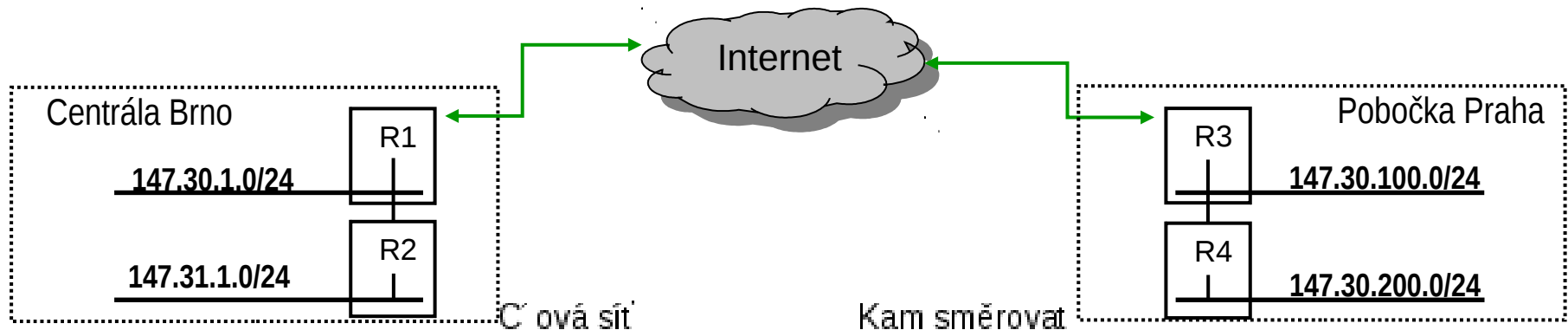
- Jak vytvořit bezpečný "intranet" bez nákladné pronajaté linky?
 - Odpověď: Použít **VIRTUÁLNÍ privátní síť** (VPN)
 - VPN kombinuje bezpečnost a ochranu citlivých dat s relativně nízkou cenou přístupu na veřejný Internet
- Princip tvorby VPN
 - Připojit všechna pracoviště na globální Internet
 - Ochránit data přenášená po veřejné síti
 - Zašifrovat data
 - Přenášet data tzv. **IP tunelem**

Princip vytváření IP tunelů

- Základem je využití principu zapouzdřování (*encapsulation*)



- Adresace a směrování při využití VPN



147.30.1.0	doučit přímo cílovému stroji
147.31.1.0	doučit přímo směrovači R2
147.30.100.0	směrovat tunelem k R3
147.30.200.0	směrovat tunelem k R3
default	poslat směrovači poskytovatele Internetu

Směrovací tabulka v R1

Možné typy VPN

- VPN lze realizovat na různých úrovních ISO-OSI modelu
- L2 VPN
 - Spoje, které "tunelují" data spojové (linkové) vrstvy
 - Někdy označováno jako "virtuální drát" (pseudo-wire)
 - Vytvářejí tak "virtuální LAN"
 - Např. do datové oblasti veřejného datagramu se uloží (třeba i zašifrovaný) ethernetový rámec
 - Na přijímací straně se vyjme a použije se jako lokálně vzniklý fyzický rámec, který může nést i fyzický "broadcast" např. pro ARP
- L3 VPN
 - Spoje zajišťující přenos "privátních IP datagramů"
 - Často realizovány jako spoje mezi dvěma stroji (point-to-point)
 - To ovšem neznamená, že na koncích spoje nemohou stát směrovače zajišťující univerzálnější logické propojení
 - Efektivnější než L2 VPN, avšak obvykle bez možnosti "*LAN broadcast*"
 - Některé síťové služby jsou závislé na této metodě, např. *Server Message Block* (SMB) protokol, který Microsoft používá pro NetBIOS síť Windows potřebuje "*LAN broadcast*"

Rozšířené implementace VPN

- Komerční VPN

- Nejčastější jsou produkty fy Cisco a jejích subdodavatelů používající často vlastní („proprietární“) protokoly
 - Některé z nich přešly časem v obecné standardy

- Oblíbené VPN

- OpenVPN

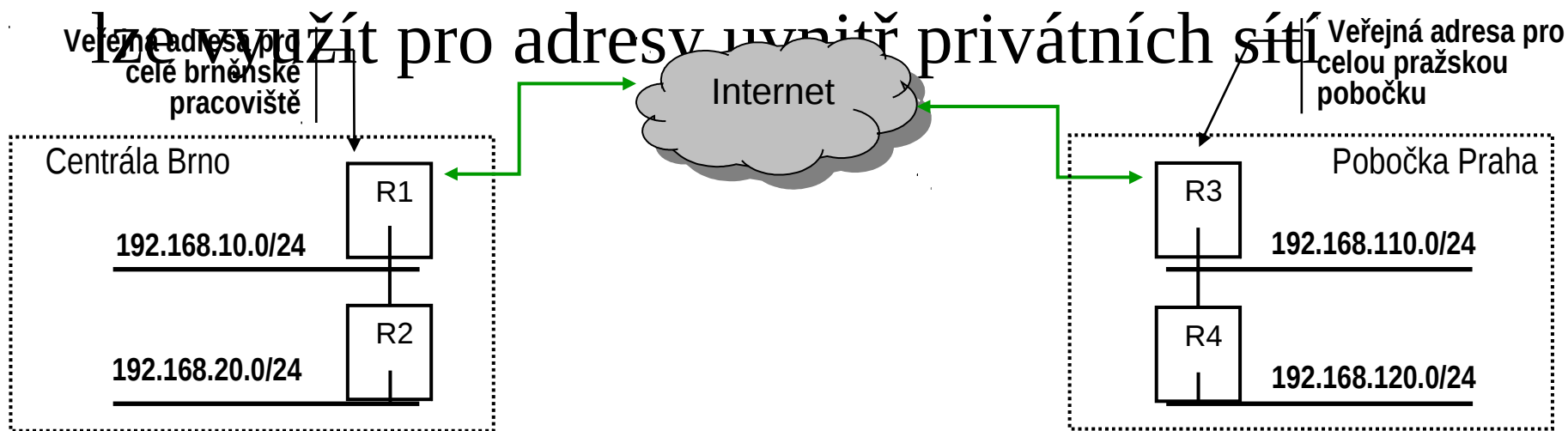
- Volně šiřitelná "open source" implementace umožňující realizaci L3 i L2 VPN
- Je k dispozici pro širokou škálu OS (Windows, Linux/Unix, existuje i klon pro Android, atd.)
- Integruje směrovače na obou koncích spoje, používá šifrované přenosy na bázi SSL/TLS (*Secure Socket Layer / Transport Level Security*)
- Transportním protokolem může být UDP i TCP

- Point-to-Point Tunneling Protocol (PPTP)

- Součástí MS Windows pod krycím názvem "*Dial-up Networking*" neboli RAS (*Remote Access Service*)
- Transportní vrstvou je TCP
- PPTP není standardizován internetovou IETF autoritou, a tak existuje řada implementací, které nejsou vzájemně plně kompatibilní
- Vedle Windows je k dispozici na Linuxech (*PoPToP*), BSD Unixech (*pppd*, *mpd* na *FreeBSD*), na Mac OS i v iPad, iPod a iPhone

Šetření IP adresami

- V našem příkladu VPN se používaly lokální sítě IP adresy z veřejného adresního prostoru
 - je to **zbytečné** a dokonce částečně i **nebezpečné**
 - Privátní adresní rozsahy
 - 10.0.0.0 – 10.255.255.255
 - 172.16.0.0 – 172.31.255.255
 - 192.168.0.0 - 192.168.255.255



- Směrovače R1 a R3 však musí zajistit i tzv.

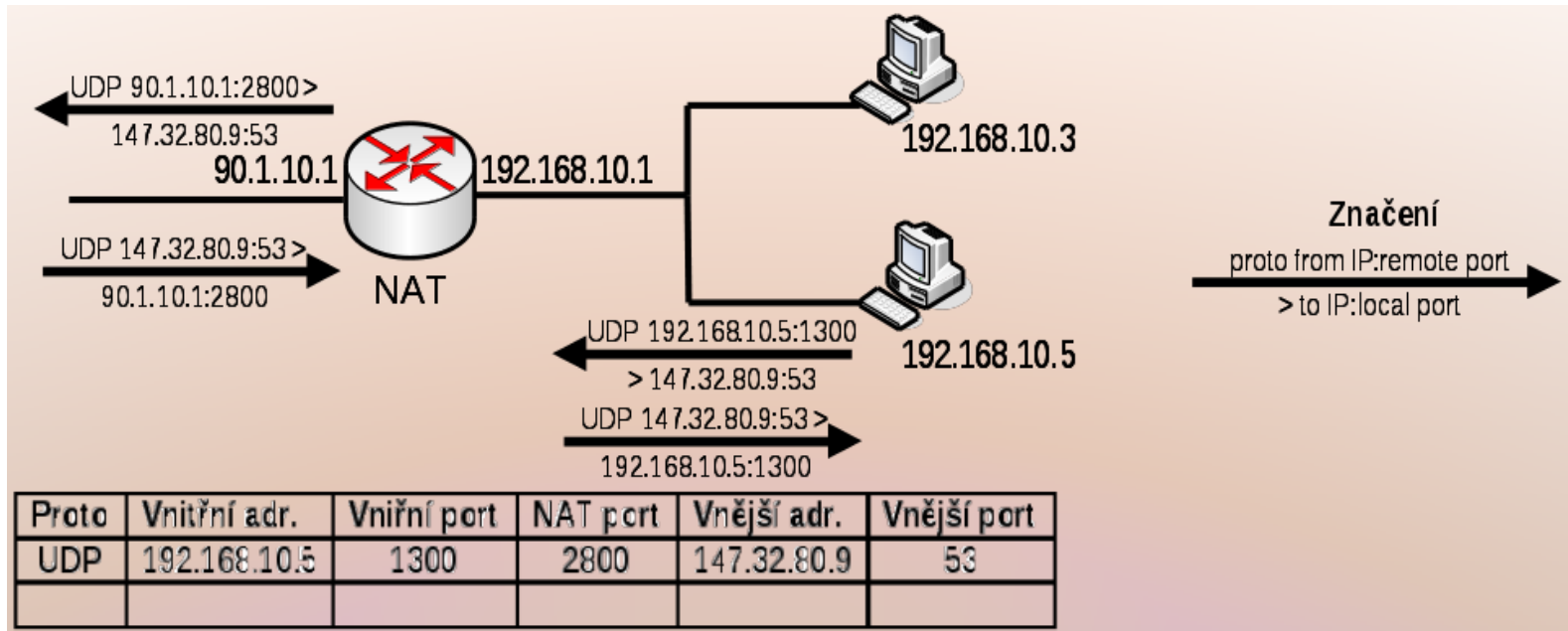
Privátní adresy a jejich překlad

- Úloha:
 - Jak lze na pracovišti s mnoha počítači zajistit přístup k veřejnému Internetu, aniž by jednotlivé stroje měly globálně platné IP adresy?
 - Řešení: Překlad privátních adres na veřejné (*Network Address Translation* = NAT)
- Požadavky na NAT
 - Překlad musí zajistit IP přístup mnoha strojů s užitím jediné veřejné IP adresy
 - Překlad musí být zcela transparentní pro obě komunikující strany
 - Implementace
 - Obvykle softwarové (nebo firmwarové) řešení
 - Nejčastěji implementováno přímo ve směrovači (*routeru*)
 - Výjimečně speciální hardware, jsou-li extrémní požadavky na rychlost
 - Detaily viz RFC 2663

Princip NAT

- **NAT upravuje záhlaví IP datagramů**
 - Uzel realizující NAT má k dispozici jednu veřejnou adresu a řadu adres privátních
 - U odchozích datagramů zamění zdrojovou privátní adresu za "svoji" veřejnou
 - U příchozích datagramů naopak cílovou adresu (ta je u příchozího datagramu veřejná) adresou privátní
 - Ale kterou z mnoha lokálních privátních adres?
 - Přesný způsob překladu je závislý na použitém IP protokolu, který je datagramem nesen
 - U UDP a TCP se překlad opírá o porty, u ICMP je to podstatně složitější
- **Překladová tabulka**
 - Dynamicky vytvářená tabulka
 - Položky se vytvářejí, když datagram odchází. Zaznamená se lokální (privátní) adresa odesílatele, cílová (vzdálená) adresa a u UDP a TCP se zaznamená **lokální port**, který se u odchozího datagramu změní
 - Při příchodu odpovědi se vyhledá v tabulce uložená vzdálená adresa a port, z čehož se odvodí lokální stroj, jemuž odpověď patří a v tabulce se najde i původní lokální port

Překladová tabulka NAT



- NAT analyzuje odchozí datagram
 - Zapamatuje si nesený protokol, adresu a port odesilatele datagramu a cílovou adresu a port
 - V hlavičce datagramu přepíše zdrojovou adresu na svojí veřejnou adresou a "vymyslí si" (a zapamatuje) "nový" zdrojový port
 - Datagram odešle cílovému stroji
- Odpověď přijde na veřejnou adresu NATu a "nový" port
 - V překladové tabulce NAT tabulce vyhledá u příslušného protokolu pár (vnější adresa, "nový" port), čímž najde odpovídající vnitřní (privátní) adresu a příslušný port. Upravený datagram pošle stroji na privátní síti.
- Protože se přepisují nejen adresy, ale i porty, bývá takový systém někdy označován jako NAPT (*Network Address & Port Translation*)

- Popsaný mechanismus funguje dobře, pokud
 - Komunikace je zahájena strojem v privátní síti
 - Pak lze založit záznam v překladové tabulce
 - Cílový stroj odpoví
 - Může se stát, že odchozí datagram nebo odpověď se ztratí
 - Ztratí-li se odchozí datagram, přijde zpět ICMP zpráva, např. "Destination unreachable". Naštěstí tyto zprávy nesou kopii záhlaví a počátek datové části ztraceného datagramu, takže tyto informace lze využít k vyhledání záznamu v tabulce
 - Informaci o ztrátě datagramu se musí dovědět původní odesílatel!
 - Ztratí-li se odpověď, musí se využít vhodného časové prodlevy k vymazání záznamu z tabulky
 - Situace je poněkud jednodušší u TCP, kdy v

Problémy s NAT

- Problémy jsou

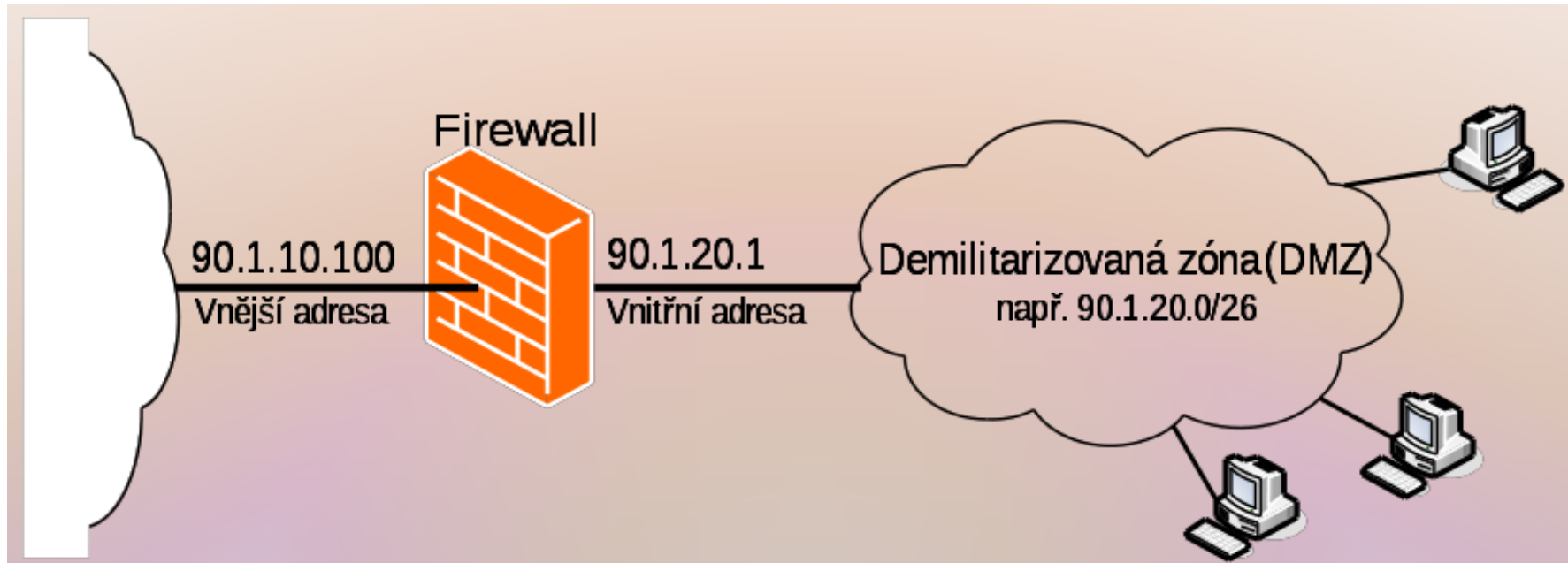
- s ICMP protokolem, který nemá porty
 - Většinou však ICMP datagramy (např. "Echo request" a "Echo reply") nesou identifikátor použitelný k orientaci v překladové tabulce
- Je-li za NATem veřejně dostupný server,
 - např. web server poslouchající na privátní adrese a veřejném portu 192.168.1.3:80

pak se externí klient připojuje na veřejnou adresu NAT a daný port

- V překladové tabulce pak bude statický záznam, který přesměruje příchozí TCP datagramy vedoucí na veřejný port (např. 80) na odpovídající privátní adresu (např. 192.168.1.3)
- Problém je v tom, že takový server může být jen jediný; v tabulce nelze mít více záznamů s tímž cílovým portem
- Mnohdy potřebujeme mít za NATem více např. webových serverů
 - např. měřicí přístroje s web rozhraním
- Možnosti řešení jsou pak dvě:
 - Další servery nakonfigurovat tak, aby poslouchaly na nestandardním portu (např. 8080); serverům vestavěným v přístrojích však mnohdy nelze změnit port
 - Za hlavní web server (poslouchající na portu 80) zvolit univerzální server, který umí pracovat jako tzv. *web proxy*, který prostředkuje

Firewally

- NAT analyzuje obsah datagramů
 - Proč totéž nevyužít obecněji?



- Firewall
 - Stroj, který odděluje chráněnou síť od "zbytku světa" a kontroluje (zakazuje či povoluje) přístupy
 - Zakázat lze nejen neoprávněné přístupy "dovnitř", ale lze blokovat i nedovolené přístupy ven
 - cenzura přístupu k nedovoleným serverům (často politická motivace – Čína, či etická – porno)

Typy firewallů

- Firewally jsou v principu tři typů
 1. Paketové filtry
 2. Aplikační brány
 3. Stavové paketové filtry (případně kombinované s detektory útoku)
- Paketové filtry
 - Nejjednodušší filtrující firewally (tzv. *stateless firewall*)
 - Pracují na principu analýzy záhlaví datagramů a rozhodují o přípustnosti průchodu firewallem na základě IP adres, protokolů, portů a dalších vlastností datagramů →
 - Příkladem je firewall **ipfw** ve starších verzích FreeBSD Unix
- Aplikační brány (též *Proxy firewally*)
 - Komunikace se dělí na dvě spojení
 - Vnější klient se obrací k firewallu s žádostí o službu (1. spojení); ten vytvoří nové spojení s vnitřním serverem poskytujícím žádanou službu (2. spojení)
 - Aplikační brána (firewall) přeposílá informace z jednoho spojení na druhé a na aplikační úrovni může dokonce kontrolovat přenášený obsah
 - Odtud název "aplikační brána" – viz dříve zmíněný *web proxy server*

Typy firewallů

- Stavové paketové filtry (*stateful firewall*)
 - Pracují podobně jako základní paketové filtry, avšak umožňují měnit a zapamatovávat si vnitřní stavy a realizovat tak konečný automat
 - Rozhodnutí o přípustnosti průchodu paketu je závislé nejen na paketu samotném ale též na "historii" zachycené ve stavových proměnných firewallu. Vznikají tak "dynamická pravidla" pro práci s pakety.
 - To přináší velké výhody zejména u "nespojových" protokolů (ICMP, UDP, ...)
 - Příkladem je firewall **ipfw2** v novějších verzích FreeBSD (FreeBSD 6 a výše) nebo **iptables** v Linuxu (v 2.4 a výše)
- Stavové paketové filtry s detekcí útoku
 - Integrují tzv. IDS (*Intrusion Detection System*)
 - Mohou např. detekovat následující situaci:
 - Spojení se "tváří", že jde o připojení k webovému serveru, avšak datová oblast TCP segmentu obsahuje příkazy či příznaky, které do **http** protokolu nepatří
 - Mnohdy se opírají o heuristické informace a různé signatury, takže pracují podobně jako antivirové programy

Pravidlové systémy paketových filtrů

- Paketové filtry rozhodují, co činit s přicházejícím paketem
 - Obvykle se opírají o soustavu pravidel ve tvaru
 1. podmínka → akce
 2. podmínka → akce
 - ...
 - N. podmínka → akce
 - Pravidla se vyhodnocují postupně. Jakmile je splněna příslušná podmínka, je přijato rozhodnutí a práce s posloupností pravidel končí
 - Celková efektivita jistě závisí na pořadí pravidel
- Akce paketového filtru
 - Základní obvyklé akce, o nichž se ve firewallu rozhoduje, jsou **allow** (propust'), **deny** (zakaž) a **forward ip** (přepošli na danou adresu)
- Podmínky akcí
 - Definují vlastnosti analyzovaných paketů
 - Soupis všech možných vlastností paketů je mimo naše možnosti – uvedeme jen několik ilustrativních příkladů

Pravidlové systémy paketových filtrů (2)

- Jako příklad uvedeme zápis pravidel pro `ipfw` (FreeBSD)
 - Předpokládejme, že DMZ je síť **90.1.20.0/26** a symbolicky ji označíme **dmz**
 - Zápis pravidel má tvar: **akce podmínky modifikátory**
 - allow ip from dmz to dmz**
 - dovolí neomezenou komunikaci uvnitř DMZ po libovolném IP protokolu
 - allow tcp from any to any established**
 - propustí všechny pakety navázaného TCP spojení. (Jak se pozná "navázané spojení"?) Bývá jako jedno z prvních pravidel kvůli efektivitě.
 - deny ip from any to any**
 - Poslední pravidlo, které zakazuje vše, co nebylo povoleno některým předchozím pravidlem.
 - allow udp from dmz to ns.provider.cz 53**
 - allow udp from ns.provider.cz 53 to dmz**
 - Tato **dvojice** umožní komunikaci strojů na DMZ intranetu komunikovat s DNS serverem poskytovatele připojení (**53** je DNS port). "Stateless firewall" k tomu potřebuje dvě pravidla.
 - U "stateful" firewallu stačí jedno pravidlo
 - allow udp from dmz to ns.provider.cz 53 keep-state**
 - Toto pravidlo se "uchytí" při vzniku DNS dotazu "zevnitř". Modifikátor **keep-state** způsobí, že se vytvoří dynamické pravidlo povolující **obousměrný** průchod firewallem pro tentýž protokol a tutéž dvojici IP adres. Dynamické pravidlo se po chvíli (zpravidla za 5 s) samo zruší .

To je dnes vše.

Otázky?