OKsystem

Jak projít přes NAT a neztratit se v překladu.

Ivo Rosol
ředitel vývojové divize

Agenda prezentace

- Funkce a chování NAT
- Detekce a kategorizace NAT s použitím STUN
- Technologie průchodu přes NAT
  - propichování děr, predikce čísel portů
  - reléování s využitím TURN serveru
  - interaktivní navázání spojení založené na ICE
  - NAT UPnP

IP adresy (IPv4)

IP adresa je strukturovaná číselná identifikace zařízení v IP síti. IPv4 má délku adresy 32 bitů, IPv6 128 bitů.

Vzhledem k notoricky známému nedostatku IPv4 adres (cca 4 miliardy), byla v 90. letech použita technologie sdílení části adresového prostoru mnoha zařízeními. LAN používají privátní adresy, které jsou skryty před vnějším světem a nejsou směrovatelné v Internetu.

Privátní adresy

Pro privátní adresy byly rezervovány internetovou organizací IANA 3 následující bloky síťových adres:

- 10.0.0.0/8 jedna adresa třídy A
- 172.16.0.0/12 16 adres třídy B,
- 192.168.0.0/16 256 adres třídy C

Privátní adresy lze používat bez jakékoli koordinace s IANA i bez jakékoli registrace, je ale **nutné použít NAT**.

Porty

TCP/UDP port je adresa služby na síťovém zařízení s IP adresou.

Servery poskytující služby by měly používat statické zdrojové „well known“ porty 0 – 1023 nebo registrované zdrojové porty 1024 – 49151,

Klienti by měli používat dynamické zdrojové porty 49152 (C000) – 65535 (často se nedodržuje).

NAT

- NAT je většinou realizován jako funkční komponenta routeru/brány na hranici mezi vnitřní sítí a internetem.
- NAT je v datové cestě, prochází jím všechny IP pakety v obou směrech, může měnit jejich adresní informaci v IP hlavičce ještě před tím, než jsou směrovány routerem, a může pakety ve spolupráci s routerem a firewallem i filtrovat (zahazovat).
- NAT přerušuje end-to-end IP konektivitu, přijímá pakety, modifikuje jejich hlavičky (viz dále) a posílá je dále, to vše činí transparentně.
- NAT je citlivý na směr průchodu paketů (na rozdíl od routeru), chová se jinak při průchodu zevnitř ven, než v opačném směru, podobně jako firewally.

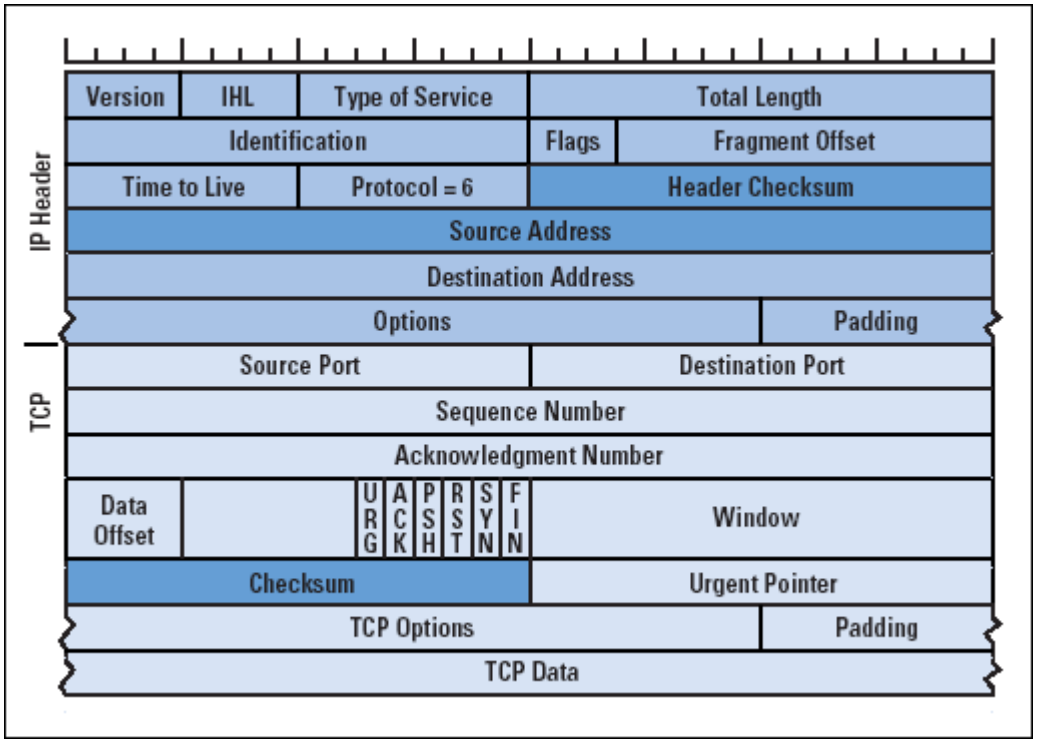
NAT

- NAT měl být původně dočasné řešení, než bude nástupce IPv4, v současné době je NAT všudypřítomný.
- NAT byl navržen jako transparentní vůči protokolu mezi koncovými komunikujícími body, bez interakce mezi koncovým zařízením a NAT.
- Pro některé aplikace a protokoly NAT funguje transparentně a bez problémů, pro jiné, jako je VoIP, je NAT překážkou.

NAT – Router - Firewall

- Pořadí zpracování příchozího IP paketu na jakékoli rozhraní je Firewall – NAT – Router – tedy nejprve se filtruje, pak se překládá a nakonec se směruje.
- NAT se chová různě pro pakety přicházející z LAN sítě, nebo z Internetu. Na rozdíl od routerů, NAT mohou měnit adresní informace v paketu před tím, než je směrován. Při průchodu z LAN do Internetu NAT mění zdrojovou adresu (a port v případě NAPT) v hlavičce IP protokolu, při průchodu z internetu do LAN mění cílovou adresu (a port v případě NAPT) v hlavičce IP protokolu. V obou případech musí přepočítat a změnit kontrolní součet v IP hlavičce i TCP hlavičce.

Změny při průchodu přes NAT



Změněné pole po průchodu NAT (z privátní do veřejné sítě) jsou označeny tmavě modře.

Při změně Source Address je nutné přepočítat i kontrolní součty. Výpočet kontrolních součtů se optimalizuje vzhledem k jejich konstrukci (nepoužívá se žádná hash funkce ale pouze součet s doplňkem do 1 tak, že se přičte rozdíl adres před a po mapování, což snižuje zatížení NAT pro tento výpočet na každém paketu.

Funkce NAT

NAT se skládá z několika mechanismů, jejichž kombinace určuje celkové chování NAT:

- mapování adres (NAT binding)
- mapování portů (Port binding)
- doba života relace a mapování (Binding timer)
- filtrování (NAT Filtering)

V dalším budou jednotlivé mechanismy blíže popsány.

„Čistý“ NAT (Basic NAT)

„Čistý NAT“, používá pouze překlad zdrojových IP adres na veřejnou adresu z množiny disponibilních veřejných adres.

Čistý NAT je použitelný pouze v případě více veřejných adres, ideálně v případě, kdy množina veřejných adres je větší než množina privátních adres v LAN.

NAT vs. NAPT

„Čistý NAT“se v kontextu TCP a UDP komunikace prakticky nepoužívá, místo toho se používá NAPT (Network Address Port Translation), aby se nevyčerpaly disponibilní IP adresy.

Mapování při překladu NAPT překládá

(*local source IP:local source port*)

na veřejnou adresu routeru a veřejný port

(*public router IP:unique port*).

Zkratka NATP se ale běžně nepoužívá, místo toho se pod pojmem NAT skrývá obojí, nejčastěji NATP.

Statický NAT

Mapuje staticky (na základě konfigurace správcem, bez časovače) obecně privátní IP/port na veřejný IP/ port.

Speciálně lze mapovat veřejné IP adresy , tj. čistý **1:1 NAT**, kdy se každá z disponibilních veřejných IP adres překládá 1:1 na určenou privátní IP adresu ,

nebo lze mapovat veřejné porty, **Port Forwarding /Full Cone NAT**, kdy je určitý port na veřejném IP adrese přeložen na určenou privátní adresu a port. Je to nejméně restriktivní NAT. Jedná se o jediný NAT, kde je NAT port trvale otevřený a umožňuje vstupní připojení z jakékoli Internetové adresy.

Příklad:

Pro přístup z internetu do vnitřní sítě je nutné vytvořit statické mapování pro každý port služby ve vnitřní síti, která má být přístupná z internetu. Standardně lze tedy pro jedinou internetovou adresu mapovat pouze jednu službu daného typu (například jediná web kamera na portu 80).

Dynamický NAT

Záznamy v mapovací tabulce se nejčastěji vytvářejí dynamicky, na základě vzniku nové relace (TCP nebo UDP), pomocí „iniciačního“ paketu z vnitřní LAN, procházejícího od určitého vnitřního počítače přes NAT do vnější sítě.

Dynamicky vytvářené záznamy jsou opatřeny časovačem, který omezuje jejich životnost.

Stavové chování NAT, relace

Proces překladu NAT může být závislý na existenci TCP/UDP/ICMP **relace** (session). Relace je chápána jako provoz (množina paketů), který je jednotně překládán na základě složeného identifikátoru relace, kterým je:

- Pro TCP/UDP: (source IP address, source TCP/UDP port, target IP address, target TCP/UDP port)
- Pro ICMP: (source IP address, ICMP query ID, target IP address)
- Ostatní: (source IP address, target IP address, IP protocol)

Poznámka: relace je definována na koncových bodech komunikace, nikoli na adrese a portu NAT, jehož nastavení je naopak závislé na relaci.

Pro daný protokol se nová relace zahajuje ve všech případech, kdy se mění alespoň jedna hodnota v pětici: *(Protokol, source IP address, source TCP/UDP port, target IP address, target TCP/UDP port)*

Tři fáze činnost NAT

- **Mapování adres a portů (Address binding)**

  Spočívá ve vytvoření mapovacího záznamu (staticky, nebo častěji dynamicky) pro účely překladu, viz dále.

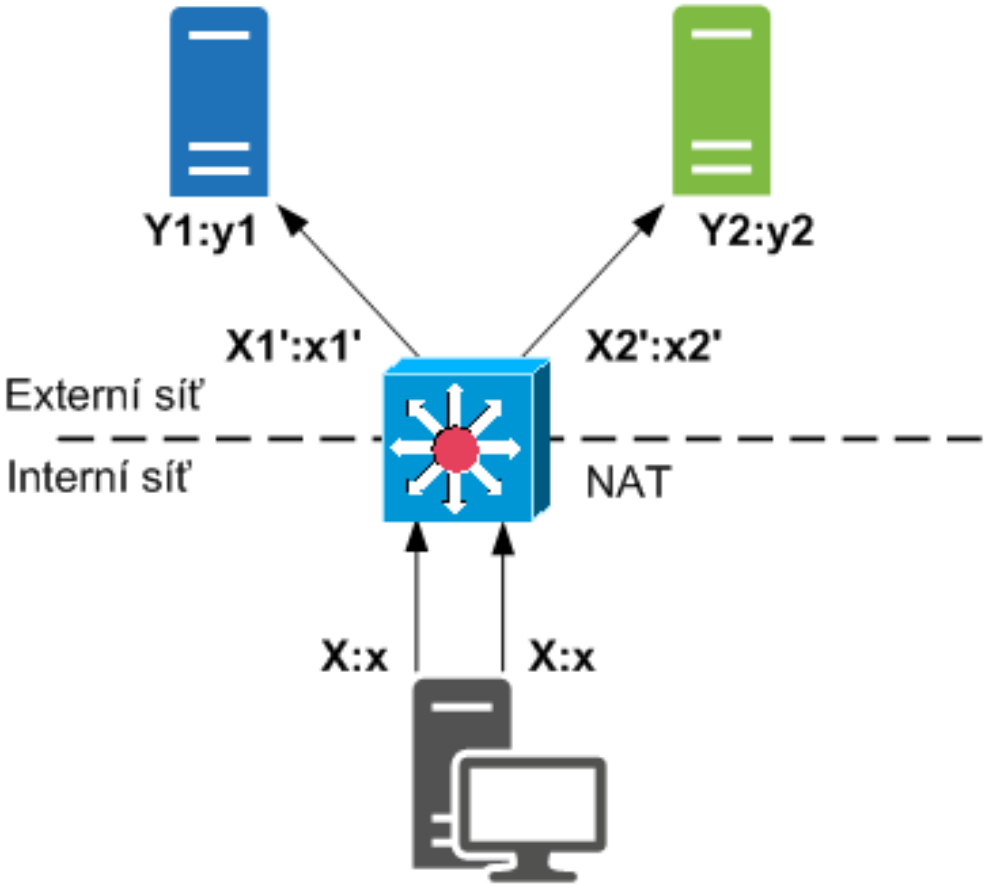
- **Addres lookup and translation**

  Jakmile je vytvořen stav pro relaci (po průchodu inicializačního paketu NATem), všechny následné pakety této relace se překládají na základě vyhledání adresy v překladové tabulce a následném překladu adres.

- **Address unbinding**

  Smazání mapovacího záznamu z překladové tabulky NAT na základě domněnky, že příslušná relace skončila (FIN, RST pro TCP nebo timeout pro UDP)

Mapování adres a portů



Pro mapování je zásadní, jak se NAT chová v okamžiku, kdy lokální PC na adrese X:x otevře další odchozí relaci (X:x=>Y2:y2) na jinou internetovou adresu, tedy zdali se pro tuto novou relaci vytvoří zcela nové mapování, nebo zdali se využije stávající mapování. Určující je tedy vztah **X1´:x1´ vůči X2´:x2´**.

Mapování (NAT binding)

a) **Nezávislé na cílové adrese** (doporučeno pro P2P)
NAT použije existující mapování, pokud lokální počítač ze stejné zdrojové adresy a portu přistupuje na jinou internetovou adresu a port. Tedy **X1´:x1´ = X2´:x2´** pro každou externí **Y:y**

b) **Závislé na cílové adrese**
NAT použije existující mapování, pouze pokud lokální počítač ze stejné zdrojové adresy a portu přistupuje na stejnou internetovou adresu ale libovolný port. Tedy **X1´:x1´ = X2´:x2´** pouze pokud **Y1 = Y2**

c) **Závislé na cílové adrese i portu**
NAT použije existující mapování, pouze pokud lokální počítač ze stejné zdrojové adresy a portu přistupuje na stejnou internetovou adresu a stejný cílový port. Tedy **X1´:x1´ = X2´:x2´** pouze pokud **Y1:y1 = Y2:y2**

Mapování více veřejných IP adres (IP address polling)

Pokud má NAT k dispozici množinu veřejných adres (public address pool, tedy více než jednu adresu), mapuje tyto adresy několika možnými způsoby:

- **Náhodné přiřazení veřejné IP adresy** z banky adres (IP address polling: Arbitrary). Lokální počítač může mít více aktivních mapování s různými veřejnými adresami. Výhodou je větší skrytí vnitřních adres.
- **Stejná veřejná IP adresa** pro všechny relace lokálního počítače (IP address polling: Paired). (Doporučeno)
- **Standardně stejná adresa, při kolizi portů může být zvolena jiná adresa** z banku, pokud je k dispozici (viz dále Port preservation)

Mapování portu
(Port binding)

Při mapování portů NAT se mohou vyskytnout následující chování:

a) Zvolení nejbližšího volného portu (No port preservation), následujícími způsoby:

- **Port parity preservation** (doporučeno)
  někdy zachovává lichost nebo sudost portů, protože některé protokoly vyšších vrstev, zvláště RTP a RTCP používají pouze liché nebo sudé porty,)
- **Zachování rozsahu portů** (doporučeno)
  někdy zachovává rozsah portů (zejména pokud je zdrojový port v rámci „well known“ portů) . Některé NAT mapují pouze na dynamické porty, jiné na registrované i dynamické, případně na všechny porty.
- **Volba následujícího portu** (port contiguity) – důvod spočívá v pravidlu, že port pro RTCP=RTP+1. Nejprve se otevře port pro RTP a poté pro RTCP.

Mapování portu
(Port preservation)

b) **snaha o zachování lokálního zdrojového portu při mapování na port na veřejném rozhraní NAT** se může zdát logická, nicméně vede na nedeterministické chování při kolizích mapování portů. Kolize se řeší následujícími způsoby**:**

- Multiplexování portů (Port multiplexing)
  Pokud 2 různé lokální počítače použijí stejný zdrojový port při komunikaci se 2 různými externími počítači, NAT zachová zdrojový port ve 2 mapováních. Pokud má NAT jedinou externí adresu, pak zvenku se jeví 2 pakety se stejnou externí adresou a stejným externím zdrojovým portem, poslané na různé cíle. Návratový paket má stejnou cílovou adresu a port veřejného rozhraní NAT a NAT určí správné mapování podle zdrojové adresy a portu návratového paketu. Pokud by 2 lokální počítače poslali paket na stejný externí počítač a použili by shodný zdrojový port, pak je nezbytné u jednoho mapování změnit (nezachovat) zdrojový port – tedy použít nedeterministické chování.
- Přiřazení jiné veřejné adresy z banky adres, pokud je k dispozici. Pokud jsou vyčerpány pro daný port všechny disponibilní veřejné adresy, vybere se jiný port (tedy port není zachován)
- Přetížení portů (Port overloading) (zakázáno)
  Některé implementace NAT se snaží o zachování zdrojového portu při mapování za každou cenu, takže při vzniku konfliktu, jakmile jiný lokální počítač vytvoří mapování s lokálním portem, který již byl mapován, tak nové mapování přepíše stávající.

(Ne)determinismus NAT

Nedeterministické chování NAT může být způsobeno několika příčinami:

- Konflikt při mapování
  Konflikt vznikne, když jiný lokální host požaduje mapování pro stejný lokální zdrojový port na stejné veřejné adrese NAT. Nedeterministické NAT mění své chování, jakmile nastane konflikt pro existující mapování, tedy chování NAT je závislé na pořadí odchozího provozu.
- Mapování jiného, než očekávaného portu
- Pokud NAT přiřazuje při mapování jiný, než nejbližší volný port, jedná se též o nedeterministické chování.
- Další komplikace pochází ze skutečnosti, že NAT se může chovat symetricky pro protokol TCP a jinak, například full-cone pro protokol UDP.

Pro multimediální P2P aplikace se doporučuje používat NAT s deterministickým mapováním. Procento NAT s deterministickým chováním se časem zvyšuje.

Filtrování NAT

- Při průchodu paketu z interní sítě přes NAT se provede na základě mapovacího záznamu v NAT tabulce překlad zdrojové IP:portu **X:x** na **X´:x´**. Při cestě zpět to je složitější a podle typu NAT se příchozí paket nejprve filtruje, zdali jde ze správné IP adresy, nebo ze správného portu, nebo z obojího.
  Filtrování se provádí na externím rozhraní NAT, filtr určuje, které pakety se zahodí a které propustí k překladu.
  Filtrování může být konfigurovatelné.
- Na rozdíl od mapování, má filtrování bezpečnostní charakter.

Poznámka: *Zdrojová adresa internetového počítače, který zasílá paket do interní sítě, se v dalším textu označuje **Z:z**.*

Chování NAT filtů

Filtrování je prováděno v závislosti na zdrojové adrese a portu počítače ve veřejné síti **Z:z**, na rozdíl od mapování (pokud je závislé), které je prováděno v závislosti na cílové adrese a portu počítače ve veřejné síti **Y:y**.

- Nezávislé na adrese ani na portu (Endpoint independent), (doporučeno pro P2P, ale nejméně bezpečné)
  NAT neprovádí filtrování, nezahazuje příchozí pakety v závislosti na jejich zdrojové adrese. To znamená, že odeslání paketu z vnitřní sítě z X:x na jakoukoli adresu v internetu Y:y je postačující, aby libovolný paket z libovolné adresy Z:z na internetu prošel přes NAT na X:x
- Závislé na adrese (Address-Dependent), (doporučeno pro P2P)
  NAT zahodí pakety přicházející na externí rozhraní, pokud se neshoduje jejich zdrojová adresa s cílovou adresou **Z≠Y**, která je uložena jako součást relace v tabulce mapování
- Závislé na adrese i portu
  NAT zahodí pakety přicházející na externí rozhraní, pokud se neshoduje jejich zdrojová adresa a port s cílovou adresou a portem **Z:z≠Y:y**, která jsou uloženy v tabulce mapování. (nedoporučeno, sice bezpečnější, neboť útočník schovaný za dalším NAT je eliminován, ale nefungují aplikace, které přijímají pakety na více než jednom portu)

Kombinace mapování a filtru

Full Cone NAT

jediné mapování iniciované zařízením z vnitřní sítě na libovolné jediné internetové zařízení, umožňuje všem vnějším zařízením přístup na toto vnitřní zařízení. Je to nejméně restriktivní dynamický NAT.

Full Cone NAT odpovídá kombinaci:

- mapování: nezávislé na cílové adrese ani na portu
- Filtrování: nezávislé na zdrojové adrese ani na portu (zpětného paketu od cílového počítače)

Kombinace mapování a filtru

NAT s kontrolou IP adresy (Restricted Cone NAT)

Aby bylo možné přijímat pakety z internetového počítače, musím na něj nejprve otevřít NAT session paketem z interní sítě.

Paket z internetu, který by byl poslán na otevřený port NATu z jiné internetové adresy je odmítnut na základě filtru.

Restricted Cone NAT odpovídá kombinaci:

- mapování: nezávislé na cílové adrese
- Filtrování: závislé na zdrojové adrese (zpětného paketu od cílového počítače)

Kombinace mapování a filtru

NAT s kontrolou IP portu (Port Restricted Cone NAT)

Filtr kontroluje, zdali se zdrojový port paketu z internetu shoduje s cílovým portem předchozího paketu z vnitřní sítě.

Port Restricted Cone NAT odpovídá kombinaci:

- mapování: nezávislé na cílové adrese
- filtrování: závislé na zdrojové adrese i na zdrojovém portu (zpětného paketu od cílového počítače)

Kombinace mapování a filtru

Symetrický NAT

Na rozdíl od předchozích 3 verzí NAT, symetrický NAT vytváří nové mapování pro každou cílovou internetovou adresu. Symetrický NAT nezachovává source port LAN klienta a místo něj použije nejbližší volný, nebo náhodně zvolený port.

Symetric NAT odpovídá kombinaci:

- mapování: závislé na cílové adrese i portu
- filtrování: závislé na zdrojové adrese i na zdrojovém portu (zpětného paketu od cílového počítače)

Vlásenka
(Hairpin operation)

Některé NAT podporují tzv. Hairpin operation (česky vlásenka), kdy 2 počítače za stejným NAT (uvnitř LAN sítě) spolu mohou komunikovat prostřednictvím mapovaných rozhraní na internetové straně NAT.

Pokud počítač z lokální LAN pošle paket na veřejnou adresu a port NAT, které má mapován jiný počítač z lokální LAN (nebo dokonce ten samý počítač na své mapování) a tento paket dorazí, pak NAT podporuje hairpinning. NAT realizuje hairpin operaci tak, že kromě překladu zdrojové adresy překládá i cílovou adresu.

Hairpin je nezbytný ve víceúrovňové struktuře NAT, bohužel ne každý NAT hairpin podporuje.

Timeout a obnovení mapování

- Při vytvoření mapování je spuštěn časovač, který měří dobu bez aktivního paketu, procházejícího přes NAT. Minimální přípustná doba časovače jsou 2 minuty, doporučená doba pro UDP NAT je 5 minut, doba časovače může být konfigurovatelná.
- Většina NAT používají pouze pakety, které odcházejí z interní sítě pro udržení mapování naživu (povinné). Některé NAT mohou používat i příchozí pakety pro udržení mapování naživu (inbound refresh), to ale může umožnit externímu útočníkovi udržet mapování otevřené neomezeně dlouho (volitelné).

Je NAT bezpečnostní prvek?

Často je NAT chápán jako jednosměrný prvek, který zabraňuje navázání relací z internetu do vnitřní sítě, někteří autoři naopak jakékoli bezpečnostní funkce NAT popírají.

Pravda je někde uprostřed, NAT má následující bezpečnostní mechanismy:

- zakrytí vnitřních adres vůči vnějšímu světu.
- Směrovost NAT a filtrování paketů (závislé na implementaci nebo nastavení NAT) na základě zdrojové adresy a portu
- Časovač omezující platnost mapování

Nejdůležitější RFC pro NAT

- *RFC 2663 – NAT terminology and Considerations*
- *RFC 3022 – Traditional NAT (Basic NAT + NAPT)*
- *RFC 4787 – NAT Traversal using UDP, požadavky na NAT pro multimediální komunikaci*
- *RFC 6888 – NAT u ISP (klient dostane privátní adresu, veřejná adresa je sdílená, mezi klientem a internetem jsou 2 NAT)*

Jak projít přes NAT

Klient v lokální síti musí vědět, jak jeho adresa „vypadá z venku“, tj. jakým způsobem je přeložena NAT a jaký je typ NAT a tuto informaci musí poskytnout protistraně (například pomocí nějakého rendezvous serveru, v rámci protokolu SIP v hlavičce SDH…).

Klient dále musí zjistit, jak se NAT chová z hlediska mapování a filtrování v závislosti na koncovém bodu a jaké jsou další parametry NAT.

Na základě těchto informací klient zjistí možnosti prostupu přes NAT a měl by vybrat nejlepší variantu.

K tomuto účelu slouží řada mechanismů a protokolů, zejména STUN, TURN, ICE, FENT – IEEE protokoly pro průchod přes NAT (NAT traversal)

Mechanismus pro zjištění typu NAT

Vysoká variabilita chování NAT vedla k vytvoření protokolu a zjišťovacího mechanismu pro zjištění, zdali mezi komunikujícími stranami je jeden nebo více NAT a jak se příslušný NAT chová.

Tento mechanismus se jmenuje **STUN**

- Dříve RFC 3489 Simple Traversal of UDP through NAT.
- Nyní RFC 5389, Session Traversal Utilities for NAT

STUN

STUN je mechanismus, pomocí kterého interní STUN klient komunikuje se STUN serverem s veřejnou adresou, aby zjistil, zdali je klient za NAT, jak se NAT chová a na jakou veřejnou adresu a port přeloží privátní adresu a port klienta.

- STUN server musí mít 2 internetové adresy a 2 porty, obvykle 3478 a 3479. STUN pracuje na TCP i UDP, primární port 3478.
- STUN server může být nalezen pomocí DNS SRV záznamu pod jménem služby _stun._udp a _stun._tcp

STUN – krok 1

Proces zjištění pomocí klasického STUN (RFC 3489) je čtyřkrokový:

1. **Zjištění, zdali je UDP komunikace povolena (firewall test) a zdali je v cestě NAT**
   STUN klient pošle UDP paket na první adresu a port STUN serveru (dotaz Jaká je moje IP adresa/IP port?). STUN server vidí veřejnou IP adresu a port, za kterou je schovaný klient a tuto informaci vrátí klientovi v datové části UDP paketu. Klient porovná svojí lokální adresu s adresou v datové části UDP od STUN serveru, pokud jsou stejné, v trase není NAT, pokud jsou různé, došlo k překladu průchodem NAT.
   V odpovědi STUN server také vrátí svojí alternativní (druhou) IP adresu a port, na kterých je dosažitelný.
   Pokud se klientovi vůbec nevrátí odpověď od STUN serveru, je nejspíše blokován UDP port firewallem (odpověď by se měla vrátit vždy, nezávisle na typu NAT, který takovou odpověď ze stejné adresy a portu nikdy nefiltruje). Protože UDP neposkytuje spolehlivé doručení, mohla se komunikace také ztratit v síti a je potřeba pokus opakovat, dokud není vyčerpán maximální počet pokusů.

STUN – krok 2

2) **STUN protokol pomocí následujících dotazů se snaží zjistit typ NAT.**
   STUN klient pošle druhý paket na stejnou adresu (první adresu STUN serveru, ale nastaví flag, aby mu STUN server odpověděl z alternativní (druhé) adresy a z jiného portu. Pokud klient obdrží odpověď, pak je zřejmé, že je za **Full Cone NAT** (protože jinak by paket z jiné IP adresy i z jiného portu byl filtrován jakýmkoli jiným typem NAT).

STUN – krok 3

3) STUN klient pošle třetí paket na druhou adresu a port STUN serveru (stejně, jako v prvním paketu). Pokud klient obdrží odpověď, v jejíž datové části jsou jiné adresy a porty než v odpovědi na první paket, pak je zřejmé, že je za **symetric NAT** (protože bylo vytvořeno nové mapování v NAT pro druhou adresu STUN serveru). Pokud jsou vrácené hodnoty stejné, pak se jedná buď o Restricted Cone (IP adresa), nebo Port Restricted Cone. K určení o kterou variantu se jedná je nutný další pokusný paket.

STUN – krok 4

4) STUN klient pošle další paket na první adresu a port STUN serveru, v paketu je tentokrát nastaven flag, aby STUN server odpověděl na stejné adrese, ale na alternativním portu. Pokud odpověď projde, svědčí to o **Restricted cone**, pokud neprojde, jedná se o **Port Restricted Cone**.

Průchod přes NAT

Propichování děr (Hole Punching)

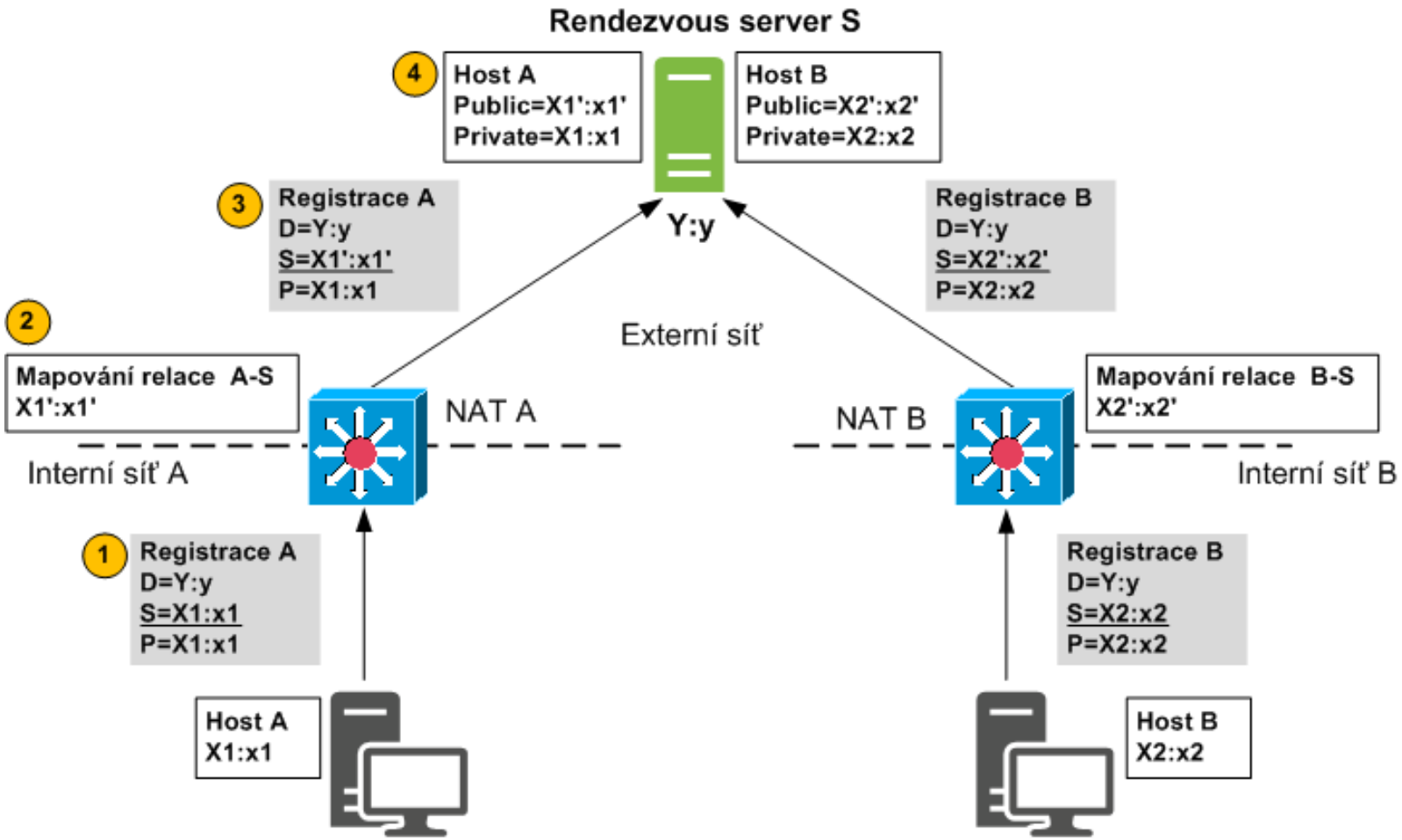
- Propichování děr umožňuje dvěma klientům sestavit přímé P2P UDP spojení za pomoci společně známého serveru (tzv. Rendezvous Server), i v případě, že oba klienti jsou umístěny za NAT.
- Přibližně 85% NAT podporuje (lépe řečeno nezabraňuje použít) technologii propichování děr pro protokol UDP (pro TCP je toto procento nižší), tato procenta se v průběhu času zvyšují.
- Propichování děr je efektivní metoda pro komunikaci, která (v protikladu k svému názvu) nevytváří bezpečnostní díry. Propichování děr skrze NAT z vnitřní sítě klientem, který hodlá komunikovat přímo s protějškem za NAT (zpravidla za dalším NAT v jiné privátní síti), lze chápat jako „signál pro NAT“, že je P2P komunikace vyžádaná interním klientem a tudíž by měla být NATem akceptována.

Registrace k Rendezvous serveru

Předpokladem pro propichování děr je konfigurace, při které mají oba P2P klienti známý (například nastavený v konfiguraci) Rendezvous Server, ke kterému oba navázali aktivní (odchozí) UDP relaci.

- Když se každý klient registruje k Rendezvous serveru, zaregistruje u něj svoje logické jméno (nespecifikováno jaké) a dále dvojici lokální IP:lokální port (privátní přístupový bod) a reflexivní IP:reflexivní port (veřejný přístupový bod). Privátní přístupový bod obdrží server v datové části klientské registrační zprávy, veřejný bod server získá ze zdrojové adresy registrační zprávy, která je po průchodu NAT přeložena na reflexivní adresu klienta.

Registrace


Rendezvous server S
4
Host A
Public=X1':x1'
Private=X1:x1
Host B
Public=X2':x2'
Private=X2:x2
3
Registrace A
D=Y:y
S=X1':x1'
P=X1:x1
Y:y
Registrace B
D=Y:y
S=X2':x2'
P=X2:x2
Externí síť
2
Mapování relace A-S
X1':x1'
NAT A
NAT B
Mapování relace B-S
X2':x2'
Interní síť A
Interní síť B
1
Registrace A
D=Y:y
S=X1:x1
P=X1:x1
Registrace B
D=Y:y
S=X2:x2
P=X2:x2
Host A
X1:x1
Host B
X2:x2

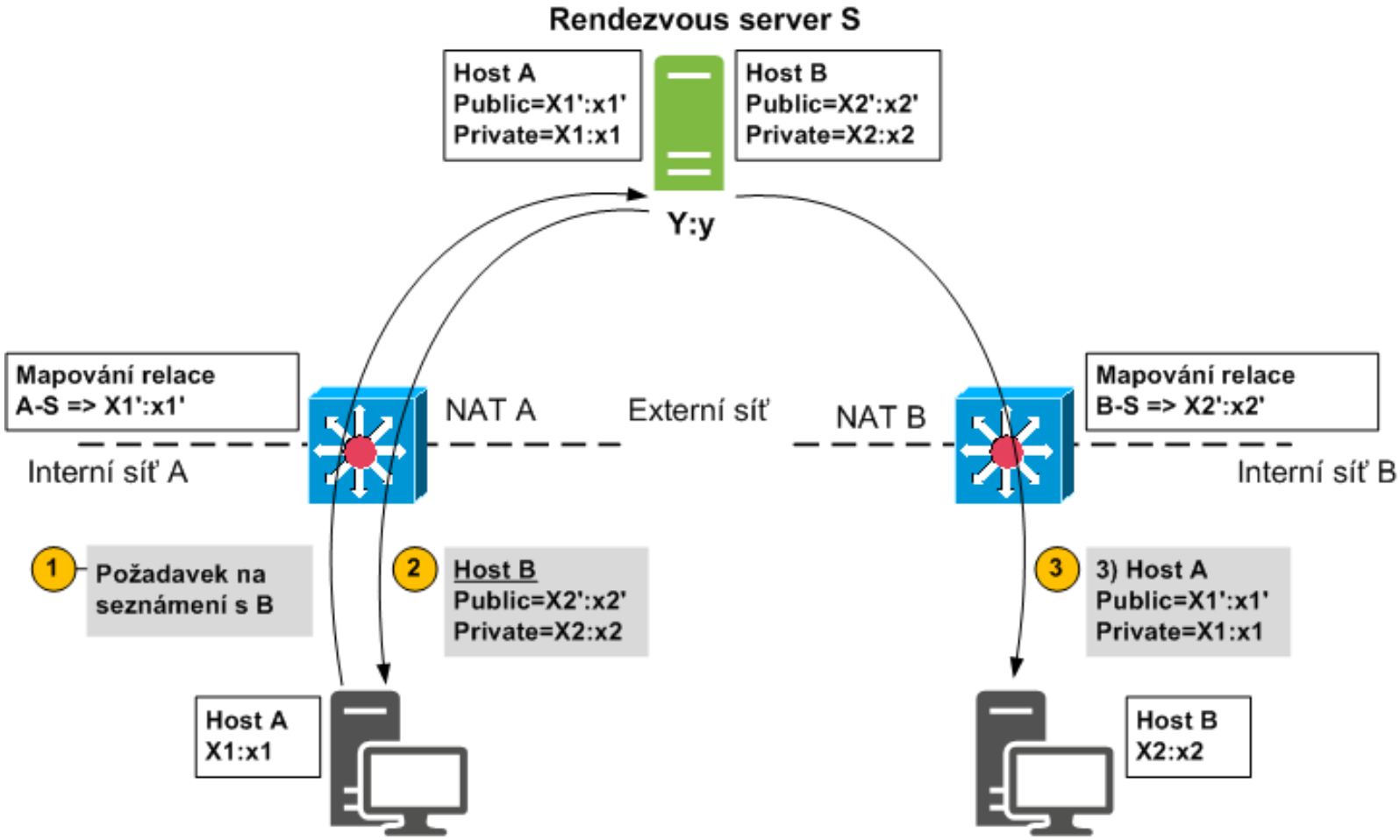
Seznámení A s B

Zpočátku A neví, jak komunikovat s B (nezná žádnou jeho adresu, pouze logické jméno, registrované na serveru S).

1) A zašle serveru S zprávu s požadavkem na seznámení s B.
2) S odpoví A zprávou, obsahující v datové části veřejný a privátní přístupový bod B.
3) S zašle B zprávu s požadavkem na pokus o přímé spojení s A (connection message). S k odeslání zprávy použije otevřenou UDP relaci s B, kterou B dříve otevřel registrační zprávou.

A i B od tohoto okamžiku vzájemně znají přístupové body protistrany.

Seznámení A s B



OKsystem

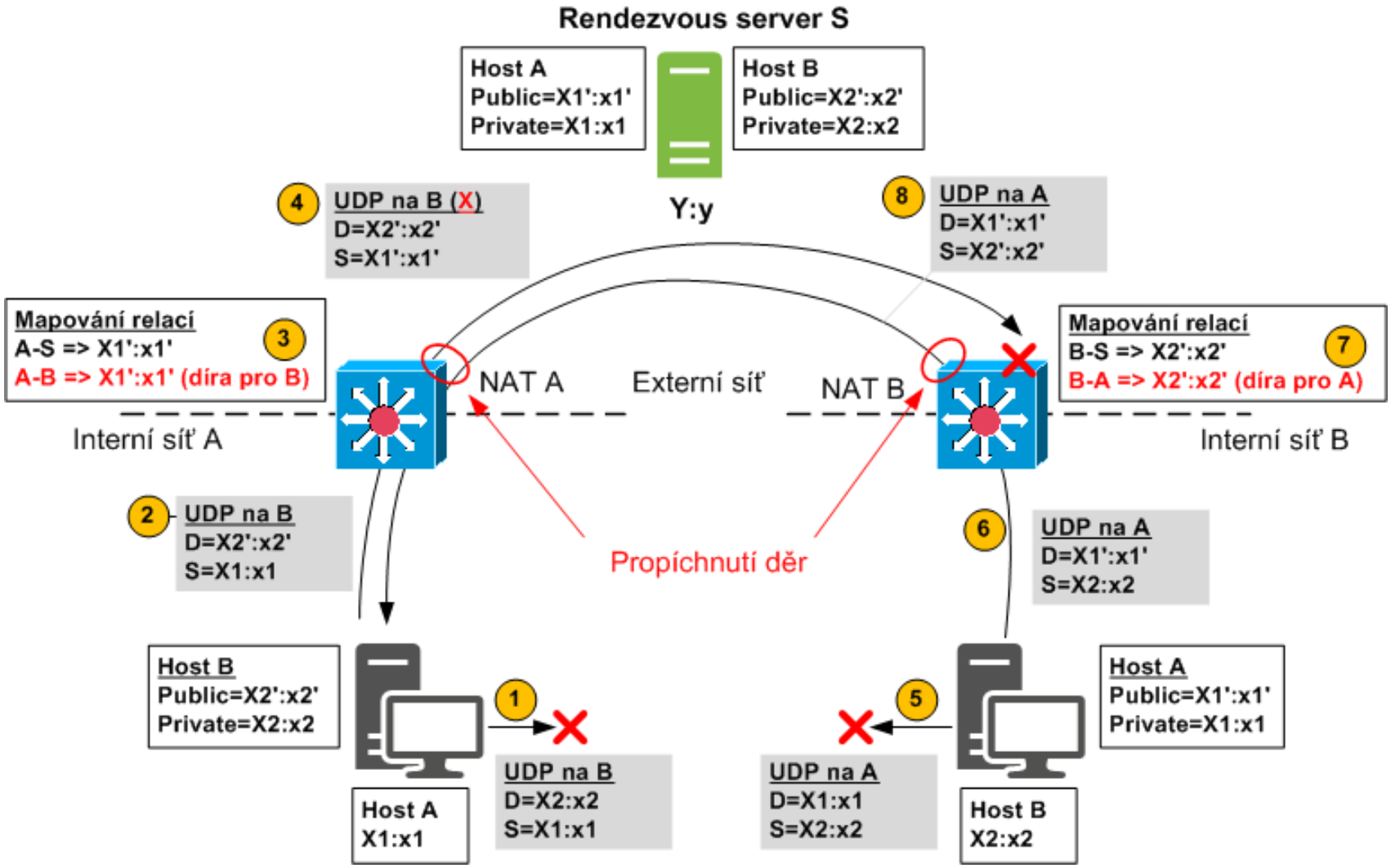
Propíchnutí děr v NAT

A i B se pokusí zaslat UDP paket protistraně na privátní i veřejný přístupový bod. Časování není podstatné, například A odešle první.

- 1)+2) Jakmile A obdrží od S zprávu s přístupovými body B, odešle A UDP pakety na oba přístupové body (A neví, zdali je B ve stejné privátní síti jako A, nebo v jiné, to není patrné ze skutečnosti, že od S přijatý privátní a veřejný přístupový bod B jsou různé). Pakety odeslané na privátní přístupový bod dojdou na špatný lokální počítač, nebo na žádný počítač (aplikace musí použít vhodnou autentizaci zpráv).
- 3)+4) Zpráva odeslaná z A ze stejného lokálního přístupového bodu, jako A odesílala serveru S požadavek na seznámení s B, na veřejný přístupový bod B při průchodu NAT (v síti A) otevře novou relaci A-B, která tvoří díru pro následující pakety z B do A za předpokladu, že NAT použije mapování nezávislé na cílové adrese. Tato nová relace A-B (označená na obrázku červeně) použije existující mapování pro A-S Tato zpráva ale neprojde NAT v síti B, neboť zde dosud neexistuje otevřená relace B-A a filtrovací funkce NAT v síti B zprávu považuje za nevyžádanou vstupní zprávu.
- 5)+6) B na základě „connection message“ přijaté od S odešle UDP pakety na oba přístupové body A (UDP na privátní přístupový bod A selže)
- 7)+8) NAT v síti B otevře novou relaci B-A, která tvoří díru pro následné pakety od A. UDP zaslaný na veřejný přístupový bod A projde propíchnutou dírou v NAT v síti A.

A i B mají od tohoto okamžiku otevřenou cestu pro oboustrannou přímou P2P komunikaci přes oba NAT.

Propíchnutí děr v NAT

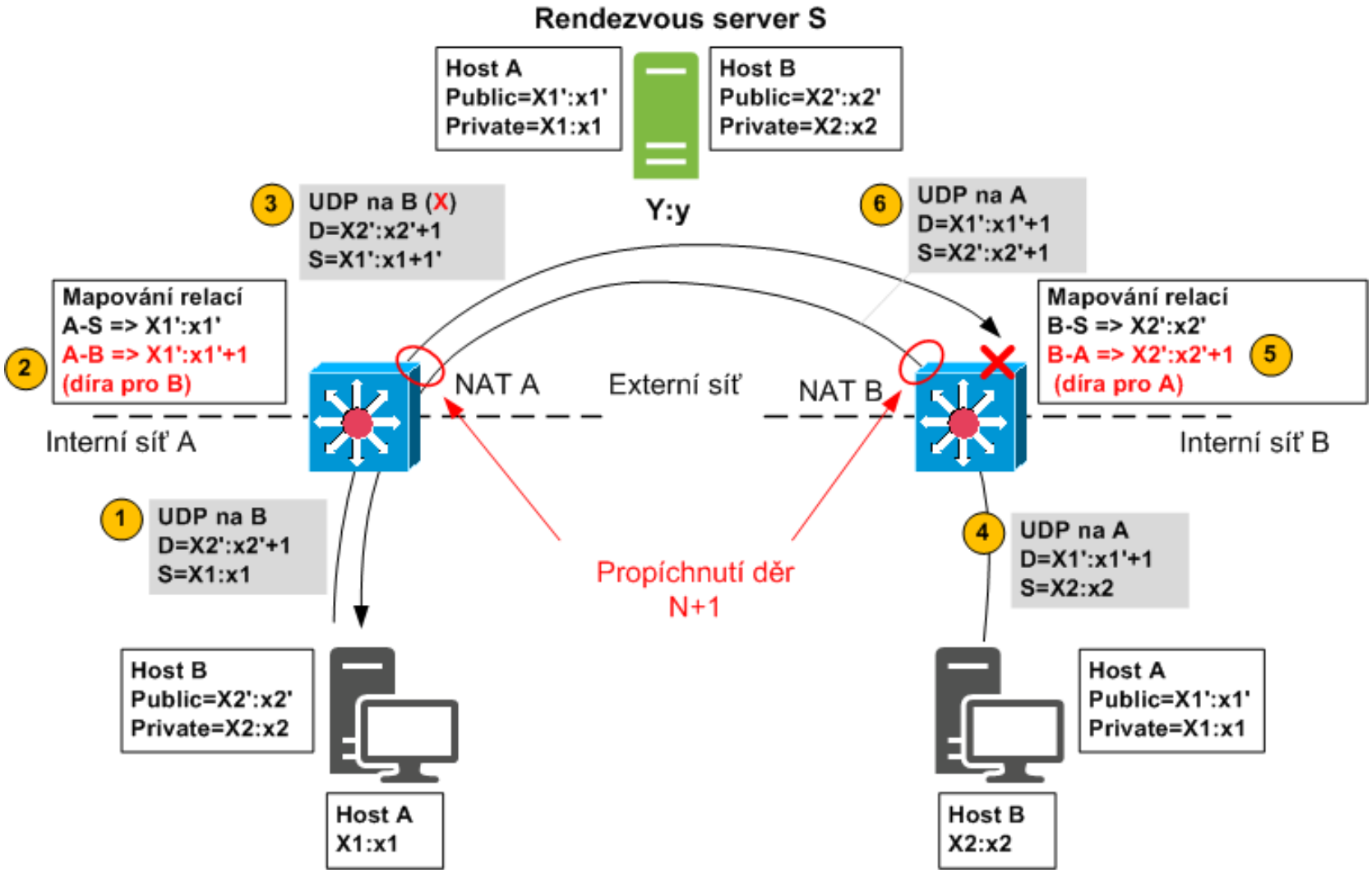


OKsystem

Predikce čísel UDP portů

- Pokud jeden nebo oba NAT používají mapování (portů) závislé na koncovém bodu, je naděje, že NAT alespoň alokuje čísla portů sekvenčně. Lze pak použít N+1 techniku, pokud uplyne pouze krátká doba mezi registrací k rendezvous serveru a následným pokusem o „strefení se“ do díry s číslem portu N+1.
- Tato technika selhává pro víceúrovňový NAT a též v případě, kdy jiný klient vytvoří nový průchod přes NAT mezi časem registrace a zasláním prvního paketu přímo mezi A a B

Predikce čísel UDP portů



OKsystem

TURN
Traversal Using Relays around NAT

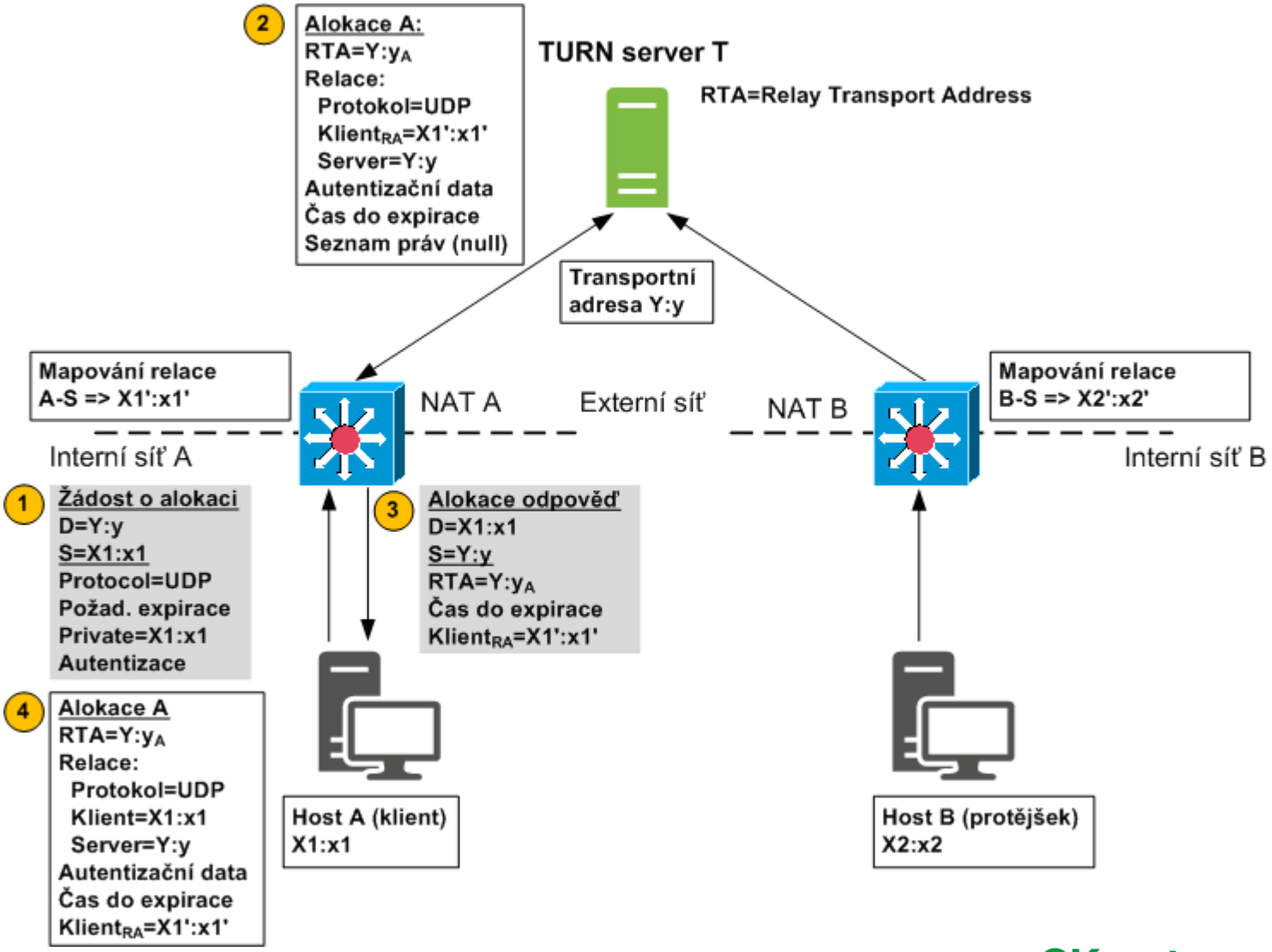
TURN je rozšíření STUN. Pokud klient detekuje pomocí STUN oboustranně symetrický NAT, tak použije TURN jako reléovou stanici (relay station = přepřahací stanice pro koně).

- TURN definuje protokol, který umožňuje klientům za NAT požádat TURN server, aby přeposílal pakety mezi klientem A a peer B. K tomuto účelu TURN server alokuje na základě požadavku klienta A „Relayed Transport Address“ (dále též RTA), na kterou protějšek B zasílá UDP pakety, které server zabalí do TURN zpráv a přepošle A.
- TURN protokol poskytuje řízení pro přeposílání paketů pro aplikace. Přeposílání je ovšem neefektivní vzhledem k šířce pásma (pro TURN server) a latenci (pro koncové zařízení). TURN je proto používaný jako poslední východisko, pokud nelze použít jiný mechanismus.

TURN - Alokace

Cílem Alokace je získání Relayed Transport Address, v podstatě se jedná o alokaci komunikačního portu na TURN serveru, která umožní datovou UDP komunikaci s klientem.

TURN - Alokace



OKsystem

TURN - Autentizace

TURN server vyžaduje autentizaci pomocí dlouhodobě platného jména a hesla, dle RFC5389 (nový STUN). Autentizace zaručuje původ i integritu zpráv (ne indikací, které jsou vždy bez odpovědí).

1. Proces autentizace začíná při požadavku klienta na autentizaci, který slouží pouze proto, aby TURN server vždy odpověděl chybou (401 neautorizováno) a vrátil v této chybové odpovědi REALM (kontext pro username) a NONCE (náhodnou autentizační výzvu).
2. Klient vytvoří nový požadavek na alokaci, tentokrát již autentizovaný na základě jména a hesla známého klientovi a na základě obdržené autentizační výzvy. Autentizační data jsou vytvořena pomocí HMAC-SHA1(obsah zprávy, včetně hodnoty NONCE a hesla).
3. Server kontroluje autentizaci přijatých zpráv a sám autentizuje odpovědi. Všechny autentizace v rámci dané alokace probíhají se stejnou hodnotou NONCE, kterou server zaslal v původní chybové zprávě.

Poznámka: HMAC-SHA1= SHA1(klíč||SHA1(klíč||data)), klíč = MD5(username:realm:password).

TURN- Nastavení oprávnění

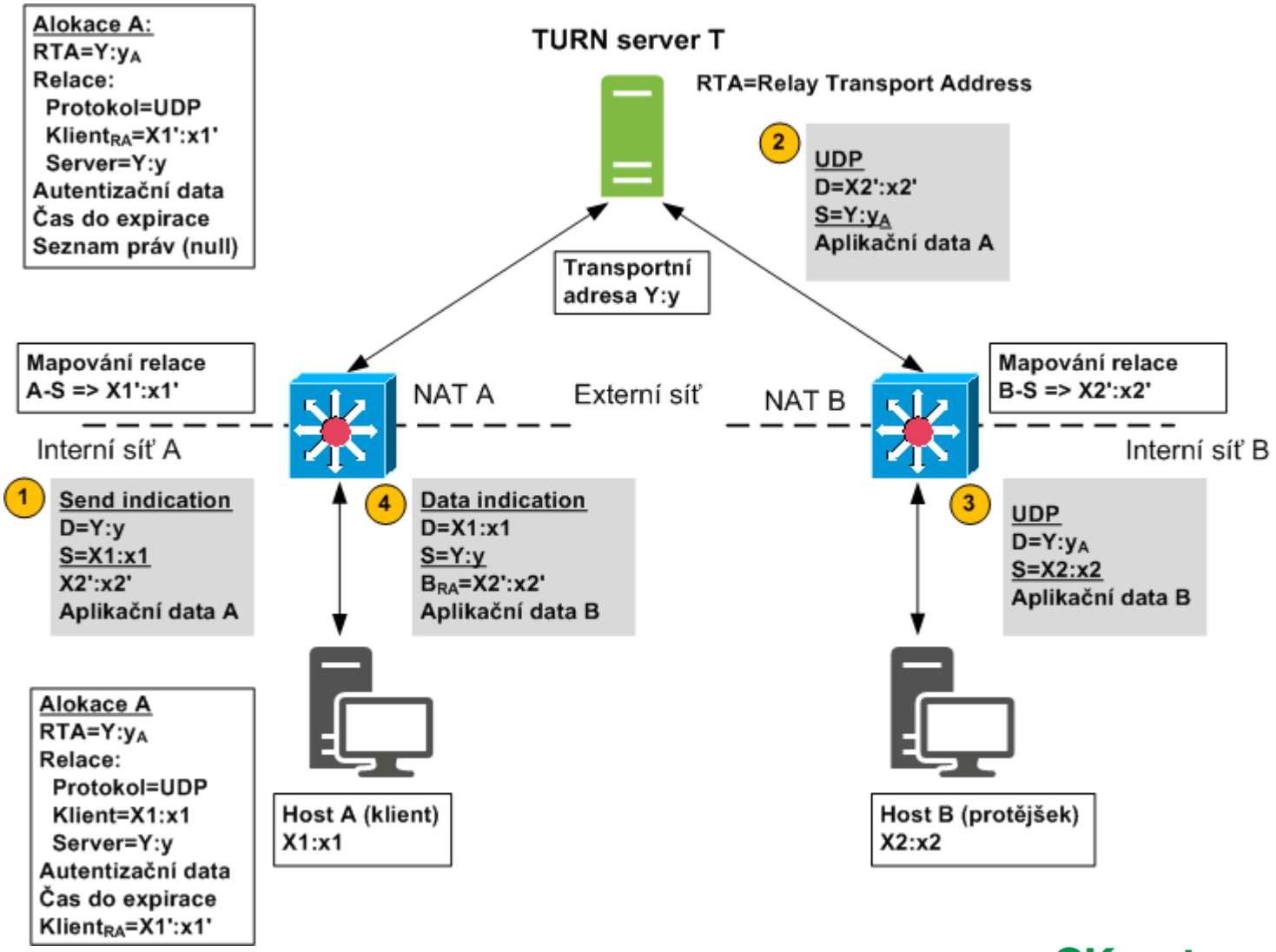
- Každá alokace vytvořená na serveru obsahuje 0 (při vytvoření) nebo několik oprávnění. Každé oprávnění je tvořeno dvojicí (IP adresa, doba do exspirace IP adresy), oprávnění neobsahuje číslo portu.
- Oprávnění instaluje klient po alokaci prostřednictvím autentizované zprávy CreatePermission, tato zpráva rovněž obnovuje hodnotu exspirace oprávnění (standardně 300 sec.).
- Jakmile oprávnění existuje, protějšky s uvedenou adresou mohou posílat data TURN serveru k doručení na klienty.

TURN - Zaslání dat

Pro zaslání zpráv od A k B prostřednictvím TURN serveru musí A znát reflexivní adresu B: X2':x2', způsob získání této adresy TURN neřeší, tato adresa se získá pomocí rendezvous serveru.

- Zaslání dat od A pro B na TURN server se realizuje zprávou „Send indication“, Obsahem „Send indication“ je reflexivní adresa B a vlastní aplikační data.
- Jakmile TURN server obdrží „Send indication“, extrahuje z ní aplikační data a zašle je prostřednictvím UDP datagramu (nikoli TURN) na reflexivní adresu B, přičemž nastaví zdrojovou adresu na (Relay Transport Address) RTA=Y:y_A, která je součástí alokace pro A.
- V opačném směru B odešle UDP datagram pro A na RTA=Y:y_A, TURN server extrahuje aplikační data a konvertuje na TURN zprávu „Data indication“, která obsahuje aplikační data (XOR) a reflexivní adresu B X2':x2'.

TURN - Zaslání dat



OKsystem

Otočení směru spojení
Connection Reversal

- Technika omezeného přeposílání pomocí reléového serveru je užitečná i v případě, že klient A je umístěn za NAT, protějšek B má globálně směrovatelnou internetovou adresu, ale navázat spojení je nutné z internetu přes NAT do privátní sítě. Předpokladem je, že A i B jsou registrováni a mají otevřenou relaci s rendezvous serverem.
- Použije se pouze omezené přeposílání, kdy server zprostředkuje přeposlání požadavku B na A, v rámci kterého si B vyžádá navázání spojení ze strany A. Tím A vytvoří průchod přes NAT, který B může využít, pokud NAT používá mapování nezávislé na koncovém bodu.

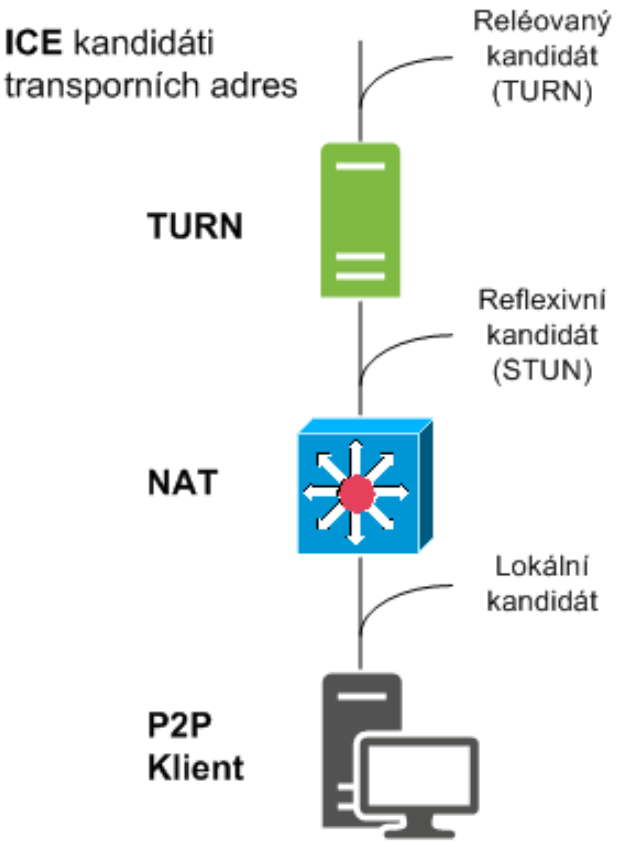
ICE

Interactive Connectivity Establishment

RFC5245

ICE zaručuje, že pokud existuje cesta mezi dvěma klienty, tak je nalezena a pokud cest je více, je vybrána optimální. K tomuto účelu ICE kombinuje STUN a TURN protokoly a vyhodnocuje priority. Vyjednávání mezi klienty je založeno na STUN protokolu.

ICE – typy kandidátů



ICE – nalezení optimální cesty

- Jakmile klient A získá a setřídí své kandidáty transportních adres, poskytne je prostřednictvím SDP protějšku B.
- B stejným postupem jako A získá a setřídí své kandidáty (přičemž není nutné, aby STUN/TURN servery byly společné) a poskytne je A.
- A i B spáruje své a protilehlé kandidáty, každého s každým, seskupí ekvivalentní páry a setřídí.
- A se stane řídícím agentem, B se stane řízeným agentem. A i B provede test spojení podle priority, aby zjistil, který pár funguje.
- Řídící agent nominuje své kandidáty, regulérně (vyzkouší se všechny), nebo agresivně (první odsouhlasený je vybrán).

Další mechanismy

NAT UPnP

- Přestože NAT je navržen jako autonomní (nespravovatelný) mechanismus, problematika řízení průchodu přes NAT vedla k vytvoření dodatečných mechanismů pro usnadnění průchodu. Tyto mechanismy vyžadují, aby příslušné protokoly měly implementovány NAT/router, aplikace nebo operační systémy.

IGDP

- Internet Gateway Device Protocol umožňuje automaticky konfigurovat NAT Port forwarding. IGD je implementováno většinou SOHO routerů.

NAT-PMP

- NAT Port Mapping Protocol byl zaveden společností Apple, popsán je v RFC 6886. NAT-PMP pracuje nad UDP a používá port 5351. Obsahuje autentizační mechanismus, umožňuje lokálním počítačům udělat díru do firewallu prostřednictvím požadavku na mapování portu.

PCP

- Port Control Protocol je následovník NAT-PMP, umožňuje interním PC požadovat překlad adres a aplikovat požadovaná pravidla pro port forwarding.

Dotazy?

Ivo Rosol
ředitel vývojové divize
OKsystem a.s.
rosol@oksystem.cz

www.getbabel.com

www.okbase.cz

www.oksmart.cz