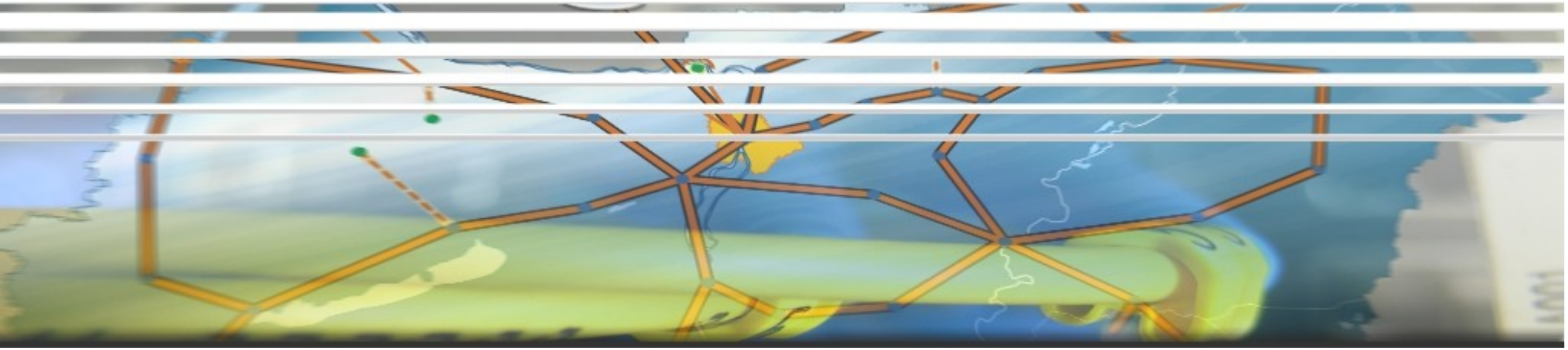
ICE(TURN/STUN)
a szabványos média tűzfalátjárási technológia, avagy hogyan tegyük Jégre(ICE) a Tűzfalat

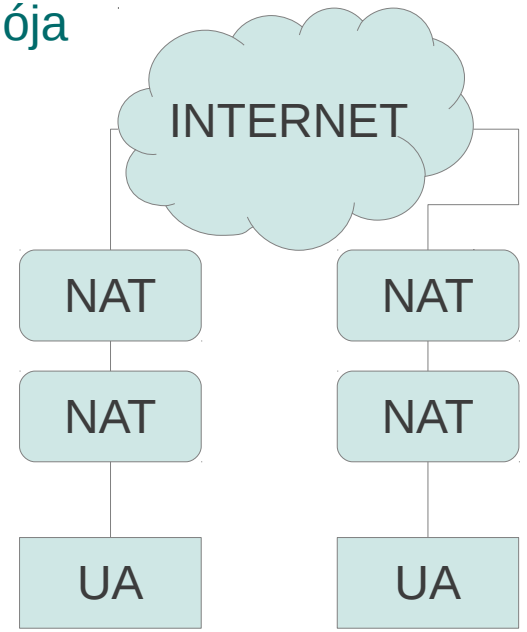


2012. április 21.
NETWORKSHOP 2012

Mészáros Mihály

Tűzfalak és a közvetlen kommunikáció(End-to-End)

- Csak média kommunikáció tűzfal átjárásról lesz szó,
  - ICE csak ezzel foglalkozik, nem foglalkozik a jelzés protokollal, arra más protokollokat használhatunk pl. SIP outbound RFC5626
- Tűzfalak (Csomagszűrés és Címfordítás) megnehezítik a kétirányú E2E kommunikációt.
- Miért fontos az E2E kommunikáció valós idejű kommunikáció esetén?
  - Késleltetés, kevesebb köztes entitás, csomagvesztés, skálázhatóság stb.
  - Intelligencia végpontokba, köztes entitások minél egyszerűbbek annál robusztusabbak
- User Agent(UA) átviteli címeinek végtelen variációja, kombinációja
  - A címfordítás (NAT), többszörös NAT, NAT több fajtája
    - Full Cone, Address Restricted Cone, Port Restricted Cone,Symmetric
  - Egy eszköz több IP cím (Multi-homed)
  - Privát és globális IPv4 cím
  - VPN, és Tunnel címek
  - IPv4 és IPv6 egyeztetés (dual-stack, ipv4 only, ipv6 only)
  - Több média stream
    - pl. videokonferencia (élőkép, hang, prezentáció)
  - Egy média stream akár több komponens (rtp,rtcp)



ICE, az univerzális megoldás

- ICE (Interactive Connectivity Establishment) RFC5245
  - Univerzális protokoll, felfedezi a környezetet és a legjobb kommunikációs lehetőséget próbálja módszeres próbákkal megkeresni.
  - Cél az E2E média használata ahol csak lehet.
  - A interneten a kommunikáció bonyolult. Így a korlátokat felfedező, figyelembe vevő, azokat ”kikerülő” megoldás is kényszerűen összetett.
    - ICE LITE kikönnyített protokoll az áttérés megkönnyítésére. Nem használható minden esetben
    - ICE FULL ennek használata javasolt, mindenütt ahol az csak lehetséges.
  - Kifejlesztése hosszú folyamat volt.
    - Több évig tartott mire draftból RFC lett (2003 február első draft, 2010 április RFC5245)
  - RFC3264 Offer-Answer modellt használja.
  - STUN/TURN szerverrel együtt ajánlott használni.
    - STUN és TURN protokoll használata, kiegészítése
  - STUN = Session Traversal Utilities for NAT – RFC5389
  - TURN = Traversal Using Relays around NAT – RFC5766

STUN/TURN

- Ajánlott felfedezés DNS SRV rekordok alapján.
  - pl.

    _stun._udp.vvc.niif.hu. 75623 IN SRV 10 0 3478 stun.vvc.niif.hu.

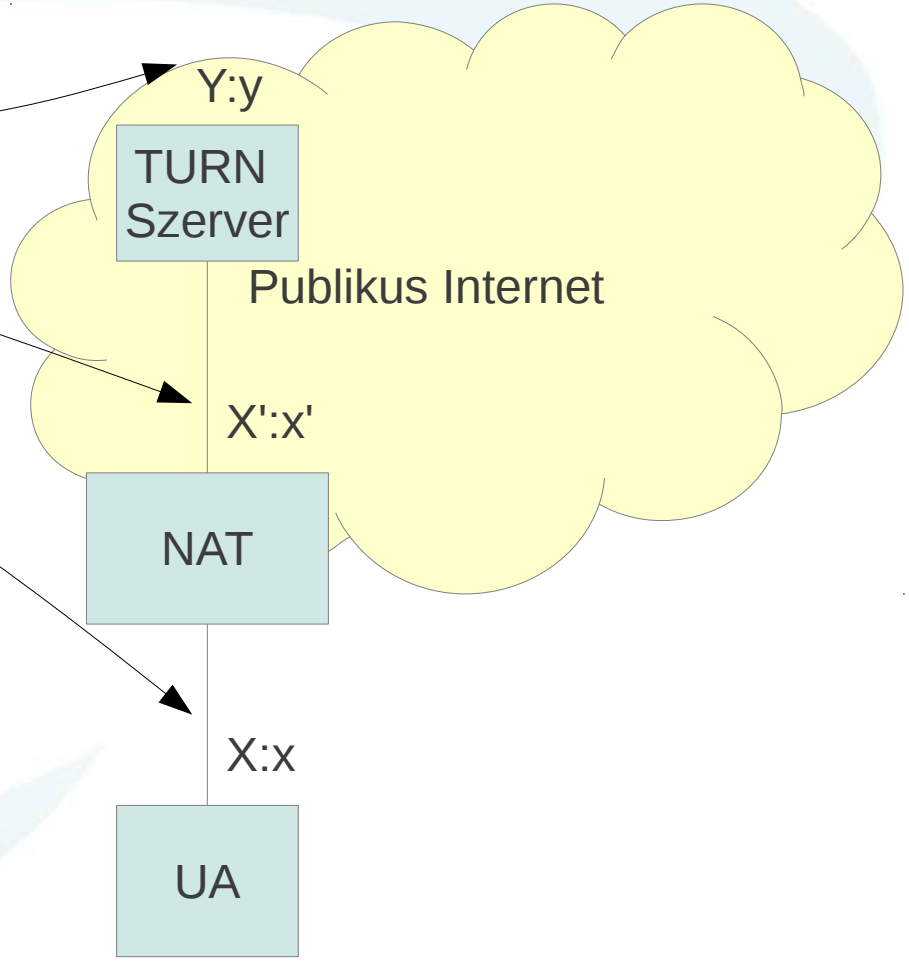
    _turn._udp.vvc.niif.hu. 86400 IN SRV 10 0 3478 hangya.vvc.niif.hu.
- STUN
  - ICE az STUN-t mint eszközt felhasználja, és definiál új STUN attribútumokat is.
  - Több módon is felhasználásra kerül, az UA és a STUN szerver között, és két UA között közvetlenül is.
  - Az újragondolt STUN, és ”klasszikus STUN”
    - RFC5389 vs. RFC3489
  - STUN Bind kérés-válasz
    - reflexive cím visszaadása
- TURN
  - TURN az STUN Relay kiegészítése, de nem STUN üzeneteket is használhat.
  - UDP, TCP, TLS támogatás
  - Az Allocate kérésre adott válaszban a Relayed és a Reflexive címet is visszakapja a kliens.
  - (RFC6156 szerint akár a kliens kérheti hogy ipv4 vagy IPv6 címet allokáljon a TURN szerver számára.)

Az ICE protokoll lépései

- Kommunikációra szóba jöhető IP címek összegyűjtése és allokációja, életben tartása
  - STUN/TURN szerver használata
  - külön minden média csatornára (hang, videó stb.)
  - ezen belül is komponensenként (rtp,rtcp)
- Priorizálás/Súlyozás, összegyűjtött potenciális kapcsolódási, átviteli címek cseréje
  - Offer-Answer modell, SDP kódolás, alapértelmezett cím
  - Párok, és ellenőrzési lista létrehozása, rendezése, duplikációk összevonása
- Kapcsolódási tesztek
  - Frozen algoritmus
  - Pár állapotok Frozen->Waiting->In-Progress->Succeeded/Failed
  - 4 utas kölcsönös ellenőrzés, két STUN binding kérés, válasz üzenetpár
  - Peer reflexive címek felfedezése
- Végső egyeztetés
  - Regular, Aggressive
- Kommunikáció, a kommunikációs csatorna életben tartása

IP címek összegyűjtése

- Átviteli cím (Candidate)
  - IP cím, port, protokoll
- Átviteli cím típusok
  - Relayed
  - Reflexive
    - Server, Peer
  - Host
- Átviteli címek alapja (base)
- Foundation (1-32 karakter)
  - Azonos típusú
  - Alapjuk közös (base ip cím)
  - Azonos STUN/TURN szerver segítségével fedeztük fel
  - Azonos protokollt használtak a cím felderítésénél
  - Frozen algoritmus használja az ICE teljesítményének növelésére



Címek prioritásának meghatározása

- Prioritás egész szám (32 bit) az ajánlott az algoritmus

  Prioritás = 2^24 * type preference +
  2^8 * local preference +
  2^0 * 256- component ID

- Preferenciák:
  - 8 bit Type preference, preferenciára ajánlott érték
    - 126 host, 110 peer reflexive, 100 server reflexive, 0 relayed
  - 16 bit Local preference
    - Azonos típusú címek preferenciája
    - VPN címnél javasolt érték 0
    - ipv6 és ipv4 közötti preferencia
  - 8 bit Component ID
    - pl. 1 rtp, 2 rtcp

Type preference	Local Preference	256-component ID

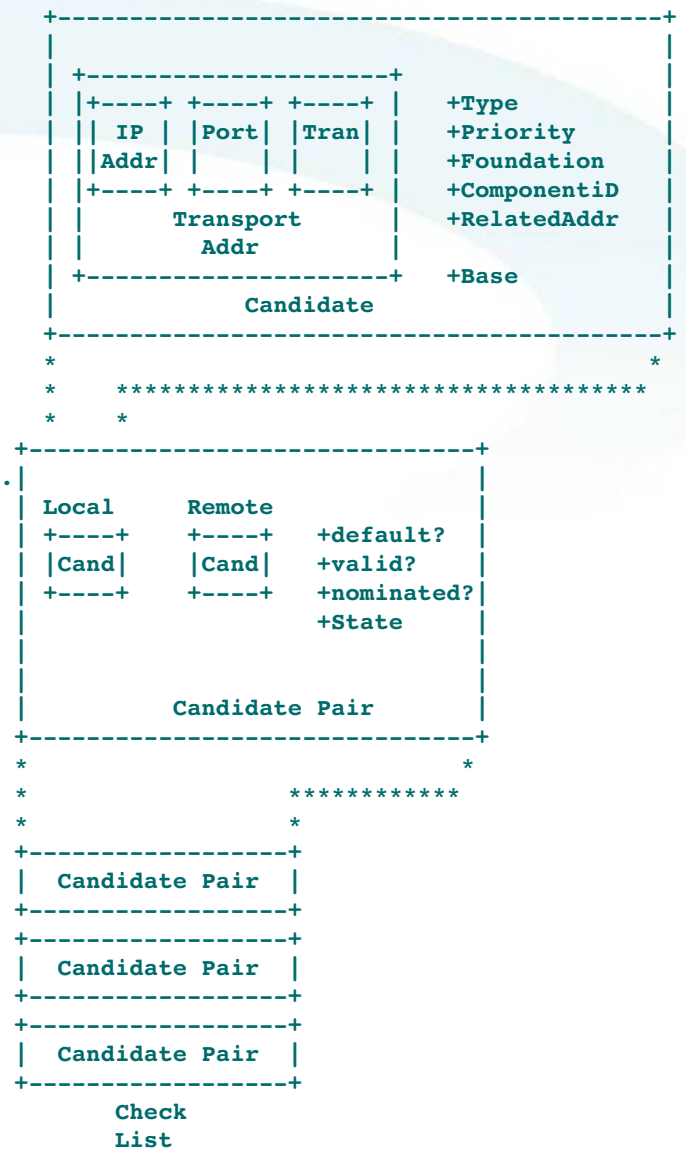
Címlista kódolása(SDP), és kicserélése

- Fontos a média sorrendje az SDP-n belül, mert, az ICE Frozen algoritmusa a legelső média, legelső komponensét fogja ”felolvasztani” legelőször.
- SDP új média attribútumok
  - ice-ufrag, ice-pwd, candidate, remote-candidate
- ICE támogatás érzékelése
  - ha default cím megtalálható a candidate-k között
- Minta SDP részlet felépítése

```
a=candidate:0a0a0a28 1 UDP 2113932031 10.10.10.40 19978 typ host
a=candidate:0a0a0a28 2 UDP 2113932030 10.10.10.40 19979 typ host
a=candidate:c0a87a01 1 UDP 2113932031 192.168.122.1 19978 typ host
a=candidate:c0a87a01 2 UDP 2113932030 192.168.122.1 19979 typ host
a=candidate:c36fc014 1 UDP 255 195.111.192.23 62377 typ relay raddr 10.10.10.40 rport 19978
a=candidate:5cf98f0c 1 UDP 1677721855 92.249.143.13 19978 typ srflx raddr 10.10.10.40 rport
19978
a=candidate:c36fc014 2 UDP 254 195.111.192.23 57606 typ relay raddr 10.10.10.40 rport 19979
a=candidate:5cf98f0c 2 UDP 1677721854 92.249.143.13 19979 typ srflx raddr 10.10.10.40 rport
19979
a=ice-ufrag:o5tW
a=ice-pwd:qr6bW6Hi3BL4X84J0gcFoE
```

Ellenőrzési lista, koncepcionális ábra (rfc5245 6.ábra)

- Pár képzés
  - azonos Component ID és azonos IP verzió
- Controlling, controlled
  - Teljes implementációk esetén az offer küldő a controlling és az answer-t küldő a controlled.
- Pár prioritás
  - Ha G a controlling, és D a controlled cím prioritása
  - Pár prioritás = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)
- Párok rendezés prioritás szerint
- A helyi reflexive címek a kiinduló(base) címekre cserélése
- Duplikációk törlése
- Így készül el a check list ahol minden pár alapértelmezett állapota ”Frozen”
- Minden média kapcsolatra külön check list
- Check list állapotok: Running, Completed, Failed
- Valid List



Frozen algoritmus

- Optimalizáció
  - A feltételezés az, hogy hasonló karakterisztikájú párok hasonló módon viselkednek.
- Algoritmus
  - kezdetben minden pár Frozen állapotban van.
  - Először az SDP legelső média stream-jének ellenőrző listájáról a legnagyobb prioritású pár-t ”olvasztja fel” (rtp utána rtcp).
  - Ha sikerül egy vizsgálat, akkor a ”hasonló”, vagyis az azonos foundation-el rendelkező más média stream-ekre is ”felolvasztja” a hasonló párokat.
    - Felolvasztás=Státusz megváltozás Frozen->Waiting
  - Ha egy Frozen pártól érkezik be STUN kérés, akkor ez felolvad. Ezt nevezzük kényszerített ellenőrzésnek (triggered check)

Párok (leegyszerűsített példa)

prioritás = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

Type preference	Local Preference	256-component ID

Prioritás	Cím	Cím
126 - 126	Host A	Host B
100 - 126	Server Reflexive A	Host B
0 - 126	Relay A	Host B
100 - 126	Host A	Server Reflexive B
100 - 100	Server Reflexive A	Server Reflexive B
0 - 100	Relay A	Server Reflexive B
0 - 126	Host A	Relay B
0 - 100	Server Reflexive A	Relay B
0 - 0	Relay A	Relay B

Párok rendezése csökkenő sorrendbe

prioritás = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

Prioritás	Cím	Cím
126 - 126	Host A	Host B
100 - 126	Server Reflexive A	Host B
100 - 126	Host A	Server Reflexive B
100 - 100	Server Reflexive A	Server Reflexive B
0 - 126	Host A	Relay B
0 - 126	Relay A	Host B
0 - 100	Relay A	Server Reflexive B
0 - 100	Server Reflexive A	Relay B
0 - 0	Relay A	Relay B

Server Reflexive => Base csere

prioritás = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

Prioritás	Cím	Cím
126 - 126	Host A	Host B
100 - 126	~~Server Reflexive A~~ => Host A	Host B
100 - 126	Host A	Server Reflexive B
100 - 100	~~Server Reflexive A~~ => Host A	Server Reflexive B
0 - 126	Host A	Relay B
0 - 126	Relay A	Host B
0 - 100	Relay A	Server Reflexive B
0 - 100	~~Server Reflexive A~~ => Host A	Relay B
0 - 0	Relay A	Relay B

Duplikáció eliminálása (pruning)

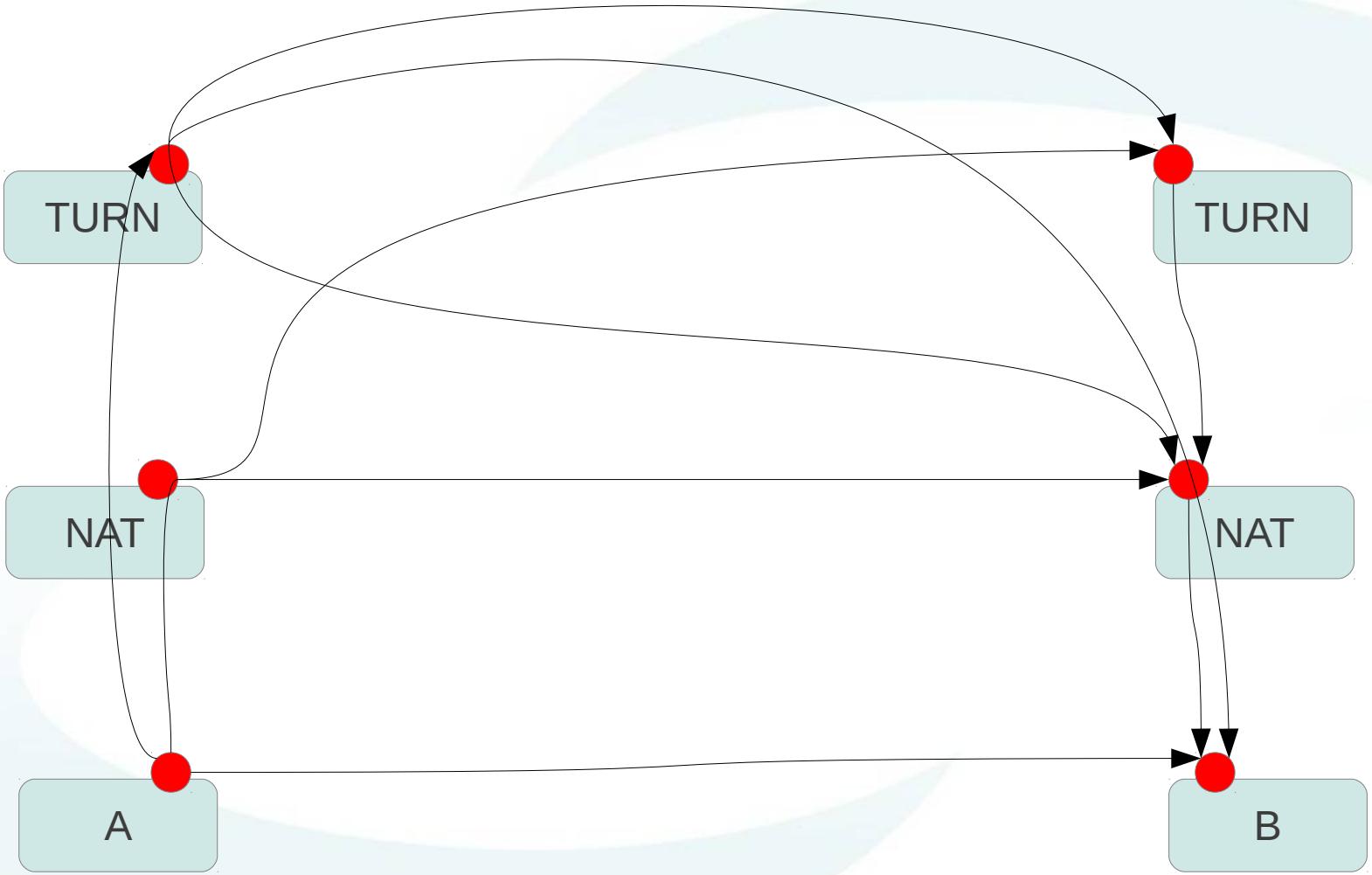
prioritás = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

Prioritás	Cím	Cím
126 - 126	Host A	Host B
~~100 - 126~~	~~Server Reflexive A => Host A~~	~~Host B~~
100 - 126	Host A	Server Reflexive B
~~100 - 100~~	~~Server Reflexive A => Host A~~	~~Server Reflexive B~~
0 - 126	Host A	Relay B
0 - 126	Relay A	Host B
0 - 100	Relay A	Server Reflexive B
~~0 - 100~~	~~Server Reflexive A => Host A~~	~~Relay B~~
0 - 0	Relay A	Relay B

Kapcsolódási tesztek

- Ellenőrzés közvetlen a tényleges média kommunikációs porton keresztül valósul meg.
  - Emiatt szét kell válogatni az STUN és a média (pl. RTP/RTCP) csomagokat.
  - Mindkét UA STUN kliens és szerver is egyben.
  - A kapcsolódási tesztek során szimmetrikus NAT esetén történik a Peer reflexive cím felfedezése
- STUN bind kérés válasz, A->B, B->A
  - Autentikáció
    - Busr:Ausr + Bpwd, Ausr:Busr + Apwd
- Ellenőrzés ütemezése
  - Ta időnként egy test (20 ms-ként)
  - Ordinary check
    - Check list
  - Triggered check
    - FIFO queue

Tesztek



Végső kiválasztás (Coordination)

- Legalább egy ellenőrzött pár szükséges, hogy sikeresen véget érjen az algoritmus
- Controlling – Controlled (master-slave)
  - STUN USE-CANDIDATE attribútum
- Kommunikációra jelölt pár (nominated)
  - Regular
    - Stabilabb, bár tovább tart
    - Valid párok közül választ valamilyen szempont szerint.
  - Aggressive
    - Minden STUN binding kérésnél használja ezt a flag-et
- Nem signaling hanem média/stun csatornát használva.
  - Ha eltér a korábban küldött alapértelmezett érték az ICE által kiválasztott értéktől, akkor új offer-answer egyeztetés küldés.
- Kommunikáció végre felépült :-)
  - Ha szükséges a kapcsolat életben tartása
  - Életbetartás/Keep-alive (rtp-no-op, rtp comfort noise, STUN indication)

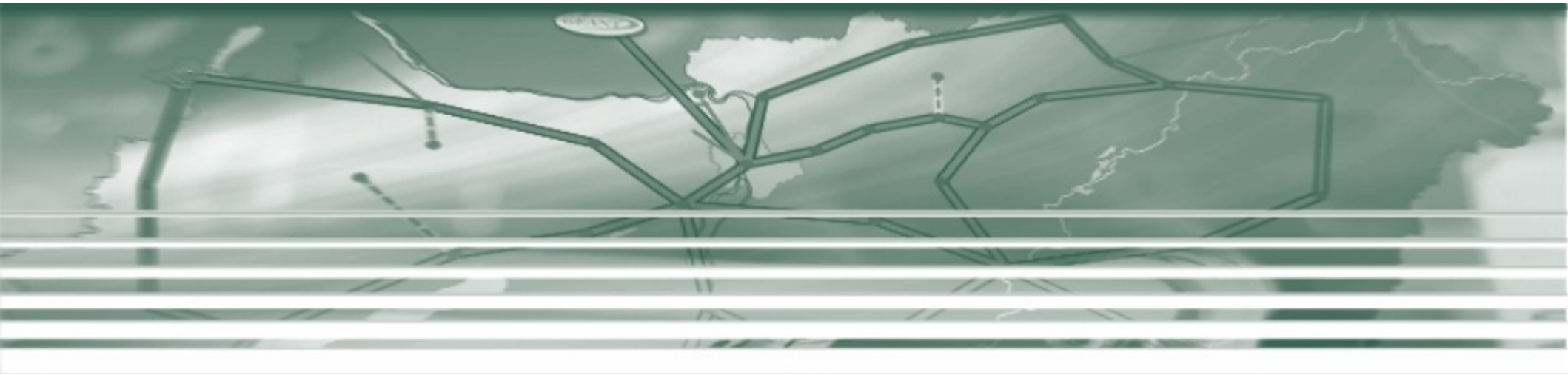
Hogyan tegyük jégre a tűzfalat? ICE(STUN/TURN)

- Az ICE(TURN/STUN) implementációk használatával
  - Implementációk libnice, pjnath, libre
- SIP-hez (RFC3261) tervezték, de más jelzés protokollal is használható
- ICE ígéretes protokollnak tűnik, előnyei:
  - Univerzális megoldás próbál adni a tűzfalátjárásra. Legrövidebb út megtalálása
  - Alkalmazkodik változó környezethez a topológia felfedezésével, és szisztematikus kapcsolódási tesztek használatával
  - E2E kommunikáció minden esetben amennyiben az lehetséges
  - Még reménytelennek tűnő esetben is képes erre
    - pl. E2E kommunikáció jöhet létre szimmetrikus NAT esetén is
  - Két azonos hálózatban NAT mögött lévő eszköz közvetlenül kommunikálhat
  - Relay használata csak legvégső esetben, csak akkor, ha már mást nem lehet tenni
  - IPv6, IPv4 protokoll egyeztetés
    - Local preference ipv4,ipv6 preferálása
    - hiba estén visszaesés
    - Az ANAT-ot az ICE nyugdíjazta, (Obsoletes: RFC 4091, 4092)

Felhasznált Irodalom

- ICE Tutorialok:
  - http://www.jdrosen.net/sip_ice.html
  - http://www.jdrosen.net/papers/ice-ietf-tutorial.pdf
  - http://www.jdrosen.net/papers/ice-basic-tutorial.pdf
  - http://sdstrowes.co.uk/talks/20081031-ice-turn-stun-tutorial.pdf
  - http://research.nokia.com/files/Strowes.pdf
  - http://www.amoocon.de/talks/93
- RFC:
  - http://tools.ietf.org/html/rfc5245
  - http://tools.ietf.org/html/rfc5389
  - http://tools.ietf.org/html/rfc5766
- Implementációk
  - http://www.creytiv.com/re.html
  - http://www.pjsip.org/pjnath/docs/html/index.htm
  - http://nice.freedesktop.org/wiki/

Köszönöm a figyelmet!



misi@niif.hu