

RSVP mérés

Készítette: Király Csaba

Célkitűzés

A mérés célja az IP alapú hálózatokban QoS biztosítására kidolgozott módszerek közül az **IntServ** architektúra és az **RSVP** protokoll megismerése (az RSVP multicast funkcióival a mérés során nem foglalkozunk).

Elméleti háttér

A mérés elvégzéséhez szükséges alapismeretek:

- IP hálózatok alapvető működése
- Szolgáltatási minőség (QoS) fogalma
- Alapvető Linux felhasználói ismeretek

A QoS biztosításának lehetőségei

Az Internet alapvető működési elve a „Best Effort” elv, mely szerint a hálózat a tőle telhető legtöbbet teszi 1-1 IP csomag továbbítása érdekében. A csomag azonban különböző okokból (pl. torlódásból adódó eldobás a routerekben vagy fejléc meghibásodása) el is veszhet vagy nagy késleltetést szenvedhet.

A csomagok eldobása és a késleltetés egyes alkalmazások (pl. IP telefon) számára kritikus lehet, míg más programok (pl. FTP letöltés) esetén nincs jelentősége (mivel a TCP megoldja az elveszett csomagok újraküldését). Fontos lenne tehát az egyes alkalmazások közötti különbségtétel, erre azonban az IP nem nyújt lehetőséget. Természetesen e nélkül is megoldható a minőség javítása:

- Egy célra dedikált IP alapú hálózat építésével: a megoldás hátránya, hogy a különböző célokra külön hálózatot kell kiépíteni és üzemeltetni, így pont az IP technológia által nyújtott integrációs lehetőségben rejlő gazdasági előnyök vesznek el. (Ezt használják pl. az IP alapú távolsági telefonszolgáltatást nyújtó cégek)
- Túlméretezéssel: azaz olyan minőségű (és költségű!) hálózat kiépítésével, mely minden alkalmazás számára biztosítja a „legszigorúbb” alkalmazás követelményeit is. Ez egyrészt nagyon költséges, másrészt technológiailag is nehezen megoldható.

Az IETF két architektúrát dolgozott ki QoS biztosítására, melyeken egy-egy munkacsoportja dolgozik(dolgozott):

- Integrált Szolgáltatások (IntServ) (IS)
- Differenciált Szolgáltatások (DiffServ) (DS)

Ezen modellek megvalósíthatóságának fontos feltétele, hogy a jelenlegi best effort szolgáltatást nyújtó architektúrát csak kis mértékben kelljen módosítani. Szintén fontos szempont mind az alkalmazások, mind a szolgáltatók szempontjából, hogy milyen jól specifikált, szerződésekben is rögzíthető garanciákat vállalhat a szolgáltató.

Bár mindkét architektúra célja QoS biztosítása, alapgondolatuk jelentősen eltér egymástól. A mérés során csak az IntServ modellel foglalkozunk.

Az IntServ modell

Az IntServ modell a jelenlegi best effort szolgáltatás mellett garantált minőségű szolgáltatások széles skáláját nyújtja. A modellben a hálózati forgalmat adatfolyamok

összességeként kezeljük. A **flyamot** azon összetartozó csomagok összességeként határozhatjuk meg, amelyek egy forrástól egy konkrét cél fele tartanak, egy alkalmazáshoz tartoznak és azonos QoS-t igényelnek. Ilyen flyam például egy videó kapcsolat egyik iránya vagy egy ftp letöltés. Egy kétirányú videokapcsolat már két flyamból áll, melyek nem feltétlenül azonos QoS-t igényelnek (pl. mpeg kódolást csak az egyik oldal tud végrehajtani, így ebben az irányban kisebb sávszélesség is elegendő).

A best effort forgalommal szemben ezekhez a garantált minőségű flyamokhoz **erőforrásokat kell lefoglalni**. A korlátozott erőforrások miatt e flyamok állapotát tárolni kell valahol, új flyamok létrehozását pedig **engedélyezni kell**, vagy erőforrások hiányában vissza kell utasítani. Mindezen feladatok a routerekre hárulnak, az eddigi állapot-nélküli routereket ki kell egészíteni néhány új funkcionális elemmel. Ezek a következők:

- **Packet classifier:** Eldönti egy beérkező csomagról, hogy melyik flyamhoz tartozik. Mint a flyam definíciójánál láttuk, ez elvégezhető a forrás és a cél IP címe és port száma, vagy egy minden csomagban szereplő azonosító alapján. A kezelés gyorsítása érdekében az azonos QoS-t igénylő flyamokat egy közös osztályba sorolhatja, így leegyszerűsödik a döntési flyamat.
- **Packet scheduler:** Feladata a csomagokat úgy továbbítani, hogy a megfelelő flyamra teljesüljenek a vállalt paraméterek.
- **Admission control:** Ellenőrzi, hogy az új flyam által igényelt paramétereket vállalni tudja-e a router a rendelkezésére álló erőforrásokkal. Természetesen a már vállalt garanciák nem sérülhetnek, így kevés erőforrás esetén vissza kell utasítani a foglalást. Fontos megjegyezni, hogy itt nem a jogosultságok ellenőrzéséről van szó.

Ahhoz, hogy a routerek ezen funkciói működhessenek, ismerniük kell a flyamokat és az azokra kért QoS-t. Az IntServ modellben ezeket az információkat nem az egyes csomagok tartalmazzák, hanem egy külön **erőforrásfoglalás** fázisban jutnak el a routerekhez. Erre a feladatra az IntServ modellben az RSVP-t (Resource ReSerVation Protocol) ajánlják.

Szolgáltatási osztályok

Eddig a szolgáltatás minőségéről csak általánosan beszéltünk, nem foglalkoztunk annak megadásával és lehetséges paramétereivel. Az IETF munkacsoport a lehetséges igényeket szolgáltatási csoportokra osztotta, ezek paraméterei határozzák meg a ténylegesen igényelt QoS-t. A best effort szolgáltatáson kívül definiált két osztály:

- Guaranteed Service (RFC 2212)
- Controlled Load Service (RFC 2211)

Ahhoz, hogy ezeket megértsük, meg kell ismernünk a flyamok leírására használt fogalmakat. Mivel az IntServ modellben a foglalások a flyamokhoz kötődnek, a foglalások leírására egy ún. **flyam-leíró**t használnak. Ennek egyrészt tartalmaznia kell a flyam azonosítását szolgáló adatokat (*FilterSpec*), másrészt a flyam kezeléséhez szükséges információkat (*FlowSpec*). Ez utóbbi több részre bontható: A szolgáltatási osztályt, a flyam tulajdonságait leíró *Tspec*-et valamint az igényelt szolgáltatást leíró *Rspec*-et tartalmazza.

Tspec: Az IntServ modell a flyam leírására a Token Bucket elvet használja. A két fő paraméter a token sebesség (*token rate*) **r** és a vödör méret (*bucket capacity*) **b**. Mindkét paraméter értékhatárait úgy állapították meg, hogy a jövőbeli hálózati technológiákat se korlátozzák. **r** értéke - amely a maximális adatsebességet határozza meg hosszútávon - másodpercenkénti 1 byte-tól 40 terabyte-ig terjedhet. A maximális löketméretet meghatározó **b** értéke 1 byte és 250 gigabyte között mozoghat. A **p** (*peak rate*) paraméter rövidávon egy löketen belül határozza meg a maximális adatsebességet. 2 további paraméter a *minimum policed unit* **m** és a *maximum pocket size* **M**. Az előbbi a nagyon kis csomagok kezeléséből adódó többletterheket kompenzálja azzal, hogy **r** számításakor minden **m**-nél kisebb továbbított csomag **m** méretűnek számít.

Rspec: Az igényelt QoS-t határozza meg, tartalma az igényelt szolgáltatási osztálytól függ. Paramétere lehet például a maximális megengedett késleltetés vagy egy adott sávszélesség.

Fontos megérteni hogy itt egy szolgáltatásról van szó, amely felfogható egy szerződésként is a szolgáltatást igénybevevő felhasználó és a szolgáltató között. Ebből a szempontból vizsgálva az egyes elemek szerepe:

- *FilterSpec*: egyértelműen definiálja, hogy melyek azok az IP csomagok, melyekre a szerződés vonatkozik
- *Tspec*: a **felhasználó** vállalása, hogy a csomagok, melyek a fenti *FilterSpec* akapján a folyamathoz tartoznak, maximum ekkora adatmennyiséget jelentenek
- *Rspec*: a **hálózat üzemeltető** vállalása, hogy a folyamathoz számára milyen minőségű hálózatot biztosít.

Természetesen (mint minden szolgáltatásnál) az üzemeltető ellenőrzi, hogy a folyamat megfelel a *Tspec*-nek, és ha nem, szankciókat alkalmazhat. Dönthet pl. úgy a *FilterSpec*-nek megfelelő, de a *Tspec* keretein kívül eső csomagoknak már a tényleges best effort körülmények között kell boldogulni.

Controlled Load Service

Ezt a szolgáltatási osztályt azokhoz az alkalmazásokhoz dolgozták ki, melyek nem igényelnek egy konkrét sávszélességet vagy maximális késleltetést, a hálózat terheltségének növekedésére azonban érzékenyen reagálnak. Ennek a szolgáltatásnak nincsenek paraméterei, az ezt igénybe vevő folyamatok számára a rendszer olyan körülményeket biztosít, **mintha best effort** szolgáltatást használnának **alacsony terheltségű hálózat mellett**.

Az alacsony terheltség ebben az összefüggésben azt jelenti, hogy a csomagok nagyon nagy százaléka eljut a célhoz, és ezek nagy részére a késleltetési idő kevéssel haladja meg a minimális késleltetési időt.

Azt, hogy ilyen szolgáltatásra szükség van, mindenki könnyen beláthatja, hiszen mindannyian használunk olyan alkalmazásokat, amelyeket a nap bizonyos szakában el sem indítunk, a hálózat túlterheltsége miatt ugyanis használhatatlanok lennének. Tehát a *Controlled Load* szolgáltatást igénybevevő alkalmazásoknak nem kell *Rspec* paramétereket megadni a foglалáshoz, csupán a *Tspec* négy értékére van szükség.

Guaranteed Service

Ez a szolgáltatás a csomagok késleltetésének maximumára nyújt garanciát, azaz minden a folyamathoz tartozó – tehát a *FlowSpec*-nek megfelelő – csomag, amely a *Tspec* keretein belül van, garantáltan megérkezik a megadott időn belül. Valószerű multimédia-alkalmazások például ezt a szolgáltatást igénylik, hiszen a csomagoknak meg kell érkezni a lejátszásuk időpontja előtt.

Egy IP csomag késleltetése két jól elkülöníthető részből áll. A fizikai átvitelből adódó késleltetés fix, míg a routerek várakozási soraiban eltöltött összidő változó hosszúságú. A sorbanállási késleltetés szabályozható a szolgáltatás két paraméterével: a vödör méretével (*bucket size* **b**) és az igényelt sávszélességgel (**R**). Könnyen belátható, hogy amíg a biztosított sávszélesség (**R**) nagyobb **r**-nél, addig a késleltetés legfeljebb **b/R** lesz.

Így tehát *Guaranteed Service* esetén nem csak a *Tspec*-et, hanem az **R**-et tartalmazó *Rspec*-et is meg kell adni.

RSVP (Resource ReSerVation Protocol)

Mint korábban említettük, a folyamat-leírók eljuttatása a kiszolgálásban résztvevő routerekhez az RSVP szerint történik. Fontos megérteni, hogy az RSVP feladata csupán az erőforrások lefoglalásának „levezénylése” a QoS paraméterek eljuttatásával a megfelelő helyekre. Sem az útvonalválasztást, sem az erőforrások lefoglalását a routerekben nem végzi el. A protokoll tehát nem értelmezi a folyamat-leíró tartalmát, csupán gondoskodik a megfelelő routerek packet scheduler és packet classifier elemeihez való eljuttatásáról. A folyamat célját

azonosító IP cím multicast cím is lehet, a következőkben azonban csak az egyszerűbb unicast esettel fogunk foglalkozni.

Az RSVP a következő hét üzenetet használja:

- Path
- Resv
- PathErr
- ResvErr
- PathTear
- ResvTear
- ResvConv

Ezek közül a *Path*-al kezdődő üzeneteket a folyam **forrása**, míg a *Resv*-vel kezdődő üzeneteket a **nyelő** küldi. Most pedig lássuk, hogyan történik a foglalás.

Első lépésben a forrás *Path* üzenetet küld a nyelő felé. Ennek két szerepe van: egyrészt routerről routerre ugorva kiépít egy utat, melyen később a folyamhoz tartozó csomagok közlekedni fognak, másrészt tudatja a nyelővel a folyam paramétereit. (Az, hogy a forrás honnan tudja, hogy a nyelőnek adatot akar majd küldeni, nem az RSVP feleadata. Egy video lejátszása előtt pl. a forrás gép kaphat egy HTTP kérést egy böngészőtől, melyből kiderül, hogy hova milyen folyamot fog majd küldeni, és ez alapján már elkezdheti a RSVP-t a *Path* üzenet elküldésével.)

Miután egy router kap egy *Path* üzenetet, az üzenetben levő *last hop* mezőből kiveszi és eltárolja az előző router címét, majd ide beírja a saját címét, és továbbküldi a *Path* üzenetet. Így amikor az üzenet eléri a nyelőhöz, rendelkezésre áll egy visszafele is bejárható út a forrástól a nyelőig.

Második lépésben a nyelő - aki már ismeri a folyam paramétereit - eldöntheti, hogy milyen szolgáltatást igényel. A *FlowSpec* és a *FilterSpec* kitöltése után egy *Resv* üzenettel kezdeményezheti a foglalást. Ezt az útvonal utolsó routerének küldi el, melynek címét már a *Path* üzenetben megismerte.

Egy router *Resv* üzenet hatására két lépést tesz: a folyam-leíró átadja az Admission Control és a Policy Control elemeknek. Az első ellenőrzi, hogy a routernek megvannak-e a megfelelő erőforrásai az igényelt QoS biztosításához, míg a második megvizsgálja, hogy az alkalmazásnak van-e joga ilyen igényelni. Amennyiben valamelyikre nemleges a válasz, a router egy *ResvErr* üzenetet küld vissza a nyelőhöz. Sikeres foglalás esetén a *Packet Classifier* a *FilterSpec*-et, a *Packet Scheduler* a *FlowSpec*-et fogja használni.

Amennyiben a két előző teszt sikeres volt, továbbküldi a *Resv* üzenetet az öt megelőző routernek.

Ha az útvonal mentén minden router végrehajtotta a foglalást, kiépül a garantált minőségű út a folyam számára. Ehhez azonban az útvonal mentén minden routernek támogatnia kell az RSVP-t, különben az útvonalon levő best effort működésű router az egész folyamra nézve megüti minden garanciát. A szolgáltatás minősége természetesen ekkor is javulhat, például ha ez a router a forrás vagy a nyelő alacsony terhelésű lokális hálózatában helyezkedik el.

A nyelő kérheti a QoS foglalás elfogadásának **explicit visszajelzését**. Ezt a *Resv* üzenetben kell jeleznie, melynek hatására a forrás egy *ResvConf* üzenettel válaszol az útvonal kiépülése után. Az üzenet megérkezésére semmilyen garancia nincs, azt egy router eldobhatja, így ezt a gyakorlatban ritkán használják.

Ahhoz, hogy a protokoll robusztus legyen, biztosítani kell hiba esetén új útvonal kiépülését, illetve a foglalás automatikus lebontását. Ezt RSVP-ben az ún. **Soft-State** mechanizmussal oldották meg: mind a *Path*, mind a *Resv* üzenetek által beállított állapotok automatikusan megszűnnek egy adott idő eltelte után, ha nem érkezik megerősítés. Így ahhoz, hogy a foglalás hosszútávon fennmaradjon, a forrásnak a *Path*, míg a nyelőnek a *Resv* üzeneteket kell periodikusan újra elküldenie.

Egy link meghibásodása esetén az útvonalválasztó algoritmus új úton fogja elküldeni a *Path* üzenetet, miközben a régi útvonal megmaradt része rövid időn belül lebomlik a régi

foglalásokkal együtt. Az új Resv üzenet már az új útvonalon fog visszalépkedni, így megpróbálhatja megtenni a szükséges foglalásokat.

Annak ellenére, hogy a nem megerősített foglalások rövid időn belül megszűnnek, az erőforrások gyorsabb felszabadítása érdekében az útvonal lebontása kérhető *PathTear* ill. *ResvTear* üzenetekkel is. *PathTear* üzenetet a folyam forrása küldhet, hatására mind az útvonal, mind a foglalások törlődnek a routerekben. A nyelő által küldhető *ResvTear* üzenet hatására csak a foglalások szűnnek meg.

Kiegészítő anyagok

A forrásokból helyi másolat is található a ~meres/RSVPmeres/doc könyvtárban. A források közül érdemes átnézni az elsőt, mely alapján a magyar nyelvű leírás készült. A CISCO RSVP leírás (3.) végén megtalálható a válasz néhány kérdésre. Aki a pontos definíciókat szereti, ezeket természetesen az RFC-kben találhatja meg.

RSVP, IntServ leírások:

1. [IBM RedBook: TCP/IP Tutorial and Technical Overview](http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/gg243376.html) (22. fejezet)
(<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/gg243376.html>)
[letöltés pdf formátumban](#) (7.7Mb !) [helyi másolat](#)
[on-line html változat](#)
2. [RSVP and Integrated Services in the Internet: a tutorial](http://citeseer.nj.nec.com/white97rsvp.html)
(<http://citeseer.nj.nec.com/white97rsvp.html>)
Megjegyzés: Későbbi tudományos munkákhoz érdemes kicsit megismerni a **citeseer** szolgáltatásait. A megadott oldalról kiindulva például könnyen találhatunk további a témáról szóló publikációkat
[letöltés pdf formátumban](#) [helyi másolat](#)
3. [CISCO RSVP leírás](http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rsvp.htm) [helyi másolat](#)
(http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rsvp.htm)

Kapcsolódó RFC-k:

4. [RFC1633: Integrated Services in the Internet Architecture: an Overview](#) [helyi másolat](#)
5. [RFC2205: Resource ReSerVation Protocol \(RSVP\)](#) [helyi másolat](#)
6. [RFC2210: The Use of RSVP with IETF Integrated Services](#) [helyi másolat](#)
7. [RFC2211: Specification of the Controlled-Load Network Element Service](#) [helyi másolat](#)

Egyéb RSVP oldalak:

8. [RSVP "homepage"](http://www.isi.edu/div7/rsvp/rsvp.html)
(<http://www.isi.edu/div7/rsvp/rsvp.html>)
9. [IETF RSVP page](http://www.ietf.org/html.charters/rsvp-charter.html)
(<http://www.ietf.org/html.charters/rsvp-charter.html>)

RSVPd: A Linux gépeken futó RSVP daemon. A mérés során az RSVP üzenetek kezelésére ezt kell használni:

man rsvpd, man rtap

[Leírás a Cisco rouerek RSVP konfigurációjáról](#)

[helyi másolat](#)

(http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt5/qcdrsvp.htm)

Iperf: A háttérforgalom és a mért forgalom generálására használt program

[leírás](http://cntic03.hit.bme.hu/~cskiraly/RSVPmeres/doc/iperf-index.html) (<http://cntic03.hit.bme.hu/~cskiraly/RSVPmeres/doc/iperf-index.html>)

A mérés során használt eszközök:

- “Forrás” PC: Linux operációs rendszer és rsvpd daemon
- “Nyelő” PC: Linux operációs rendszer és rsvpd daemon
- 2 Cisco (26xx 36xx) router
- A háttérforgalom generálásához használt PC-k: 2 PC “Forrás” és 1 PC “Nyelő” oldalon
- Az RSVP üzenetek megfigyelésére használt PC-k

Mérési konfiguráció

A laborban két 10 Mbps Ethernet hálózat található, az egyik a forrás, a másik a nyelő oldal. Mindkét hálózathoz kapcsolódik egy Cisco router, melyeket egymással egy 2 Mbps sebességű soros kábel köt össze.

A forrás (vagy a nyelő) oldalon levő Ethernet hálózaton egy további forgalomgenerálásra használható gép helyezkedik el.

Beugró kérdések

Tárolnak-e a routerek állapotinformációt az RSVP folyamatokról?

Mikor szűnik meg a routerben a tárolt állapotinformáció?

Mi történik, ha a forrás és a nyelő közötti útvonalon van egy RSVP-t nem támogató router?

Mi a soft-state szerepe az RSVP-ben?

Mire való az RSVP Path üzenete?

Mire való az RSVP Resv üzenete?

Miért használják ritkán az RSVP ResvConf üzenetet?

Mi az RSVP szerepe a QoS biztosításában?

Hogyan különbözteti meg egymástól az IP csomagokat egy RSVP képes router?

A forrás vagy a nyelő indítja a Path üzenetet?

A forrás vagy a nyelő indítja a Resv üzenetet?

Egyirányú vagy duplex foglalatot biztosít az RSVP?

A Best-effort szolgáltatási osztály mellett milyen két szolgáltatási osztályt definiál az IntServ?

Milyen garanciát nyújt a Controlled Load szolgáltatási osztály?

Milyen garanciát nyújt a Guaranteed Service szolgáltatási osztály?

Mérési feladatok

1. RSVP felépítés, forgalmi paraméterek mérése különböző háttérforgalom mellett

a, engedélyezze a CISCO routerekben az RSVP használatát, majd a Linux gépeken rsvpd segítségével foglaljon le erőforrásokat a forrás és nyelő közötti útvonalon (a mérésvezető által megadott Tspec és Rspec paraméterekkel)

Megjegyzés: természetesen az rsvpd API felülettel is rendelkezik, a mérés során nem ezt használjuk, hanem “kézzel” indítjuk a folyamatot. Ehhez az **rsvpd** programot a **-D** kapcsolóval kell elindítani, így az rsvpd-n kívül rögtön elindul az **rtap** nevű program is. Használatának leírása:

```
man rsvpd
```

```
man rtap
```

A rsvpd-nek a szomszédos RSVP-képes router IP címét is meg kell adni a -R paraméterrel.

Tipp: az RSVP session-t mindkét oldalon definiálni kell az rsvpd (rtap) programban

Tipp: ha a forrás és a nyelő közötti útvonalra nem sikerül a foglalás, próbálja meg a foglalást két azonos LAN-on levő gép között. Így a két rsvpd közvetlenül kommunikál egymással, és kiderül, hogy az rsvpd használatában vagy a CISCO routerek beállításában van-e a hiba.

b, A forrás és a nyelő közötti forgalommal és megfelelő háttérforgalom generálásával mutassa meg a foglalás hatását. A mérést végezze el kétféle különböző nagyságú háttérforgalommal. Ehhez az alábbi lépések szükségesek:

- indítson a letöltéssel azonos irányú háttérforgalmat a routereken keresztül:
Ehhez használhatja az **iperf** programot, mely a ~web/RSVPmeres/iperf könyvtárban található. Használjon TCP háttérforgalmat.

Tipp: Kliens oldalon az alábbi paramétereket érdemes használni: -P -t -i

- Mérje meg a hálózat átviteli sebességét QoS nélkül, majd QoS-el (Megjegyzés: Érdemes a háttérforgalom átviteli sebességét is mérni ellenőrzésképpen)

Tipp: ehhez szintén használhatja az iperf programot, ha a használt port megfelel a foglalásban megadottnak.

c, A b, feladatban indított háttérforgalom leállítása után indítson háttérforgalmat a forrás (vagy nyelő) oldali LAN-on. Így is mérje meg a hálózat átviteli sebességét a forrás és a nyelő gép között QoS nélkül, majd QoS-el. Magyarázza meg a c, és a b, eset közötti különbséget

2. RSVP felépítés/lebontás folyamata:

a, Dokumentálja, hogy a felépítés/lebontás során milyen RSVP üzenetek mentek a hálózaton

A, teljesíthető

B, nem teljesíthető (pl. br >2 Mbit) foglalás esetén

Az rsvpd és a CISCO routerek között a forgalmat figyelheti tcpdump-al, de az rsvpd is megfelelő mennyiségű információt ír ki ezekről (érdemes az rsvpd által generált log fájlt használni:

/var/tmp/rsvpd.log.*). A két CISCO router közötti forgalom megfigyezésére ebben a mérési konfigurációban nincs mód, ezekkel nem kell foglalkoznia.

b, Mérje meg a Path üzenet elküldésétől a megfelelő Resv üzenet megérkezéséig eltelt időt.

Tipp: Ahhoz, hogy az rsvpd rögtön Resv üzenettel válaszoljon a beérkező Path üzenetre, a megfelelően felparaméterezett "reserve" parancsot még a Path megérkezése előtt kell kiadni. Ekkor az rsvpd egy "No path information" hibaüzenetet ad, de a Path beérkezésekor rögtön a megadott Resv üzenettel fog válaszolni.

Mérési Jegyzőkönyv RSVP mérés

A mérés időpontja:

A mérésen résztvevők neve és NEPTUN fedőneve:

Mérésvezető neve:

A mérési feladatok megoldásai

1. A mérésvezetőtől kapott Tspec és Rspec értékek:

Tspec: r: b: p: m:

Rspec: osztály: paraméterek:

b,

| | leírás (paraméterek) | átvitel foglalás nélkül | átvitel foglalással |
|----------------|----------------------|----------------------------|---------------------|
| háttérforgalom | | | |
| adat | | | |

| | | | |
|----------------|--|--|--|
| háttérforgalom | | | |
| adat | | | |

c,

| | leírás (paraméterek) | átvitel foglalás nélkül | átvitel foglalással |
|----------------|----------------------|----------------------------|---------------------|
| háttérforgalom | | | |
| adat | | | |

| | | | |
|----------------|--|--|--|
| háttérforgalom | | | |
| adat | | | |

b, és c, eset közötti különbség oka:

2.

a,

| időpont | RSVP üzenet típus | UDP/IP forrás | UDP/IP cél |
|---------|-------------------|---------------|------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

b,

Eltelt idő: