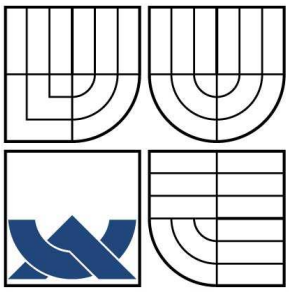
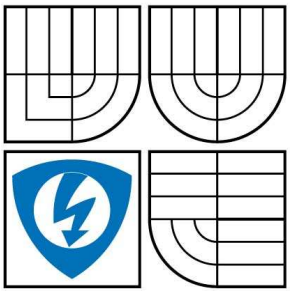


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE STANDARDU RFC 3550 V PROSTŘEDÍ NS2

RFC 3550 IMPLEMENTATION TO NS2 ENVIRONMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE BC. MICHAL HARTIG
AUTHOR

VEDOUCÍ PRÁCE ING. MILAN ŠIMEK
SUPERVISOR

BRNO 2008

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Michal Hartig

Bytem: Nový Svět 377, 51246, Harrachov

Narozen/a (datum a místo): 7.6. 1980, Jilemnice

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.

(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☑ diplomová práce
   - ☐ bakalářská práce
   - ☐ jiná práce, jejíž druh je specifikován jako.....

(dále jen VŠKP nebo dílo)

Název VŠKP: Implementace standardu rfc 3550 v prostředí NS2

Vedoucí/ školitel VŠKP: Ing. Milan Šimek

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: ……..

VŠKP odevzdal autor nabyvateli v*:

- ☑ tištěné formě – počet exemplářů 2
- ☑ elektronické formě – počet exemplářů 1

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: ……..

…….. ……

Nabyvatel Autor

ANOTACE

Cílem této práce bylo vytvořit simulační IPTV multicastový model s využitím RTP protokolu. K návrhu a simulaci bylo využito prostředí NS2, které je volně dostupné pro operační systém Linux. Pro samotnou simulaci je pak využita knihovna RTPSession, která je již součástí NS2. Zde je stále využívána norma rfc 1889, která by měla být nahrazena novou normou rfc 3550, která se již běžně používá v praxi. Práce se dále zabývá právě popisem normy rfc 3550 z roku 2003 a změnami, které přináší oproti starší normě rfc 1889 z roku 1996. Tato nová norma by měla být implementována do prostředí NS2, právě do knihovny RTPSession. Návrh simulačního multicastového modelu obsahuje jeden zdroj RTP dat a dále pak větší množství příjemců. Simulace se pak zaměřuje především na přenos RTCP–RR, SR paketů.

Klíčová slova: Multicast, ASM (Any Source Multicast), SSM (Specific Source Multicast), simulační prostředí NS2, transportní protokoly RTP (Real-time Transport Protocol), RTCP (Real-time Transport Control Protocol).

ABSTRACT

This master's thesis aimed to design the simulative IPTV multicast model based on the RTP protocol. For the design and simulation, the work uses the NS2 environment which is freely accessible for the operating systems Linux. As far as the simulation is concerned, it makes use of RTPSession library included in the NS2. It still follows RFC 1889 which should be substituted by a new RFC 3550 which already tends to be frequently used in practice. Furthermore, the thesis focuses on the description of RFC 3550 published in 2003 and it describes all the changes which were introduced since RFC 1889 published in 1996. The new standard should be implemented into the NS2 environment, specifically into the RTPSession library. The simulative multicast model-proposal embraces one RTP data source and at the same time, a great number of receptors. The simulation concentrates in particular on the transmission of RTCP-RR, SR packets.

Keywords: Multicast, ASM (Any Source Multicast), SSM (Specific Source Multicast), Network Simulator Version 2, Real-time Transport Protocol (RTP), Real-time Transport Control Protocol (RTCP).

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace standardu rfc 3550 v prostředí NS2“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Milanu Šimkovi, odbornému asistentovi Ústavu telekomunikací, za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne ....

(podpis autora)

Použité zkratky

AES	Advanced Encryption Standard
APP	Application-Defined RTCP Packet
ASM	Any Source Multicast
ATM	Asynchronous Transfer Mode
CNAME	Canonical End-Point Identifier SDES Item
CSRC	Contributing Source
DES	Data Encryption Standard
DVMRP	Distance Vector Multicast Routing Protocol
GNU	General Public License
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPsec	IP security
MOSPF	Multicast Open Shortest Path First
MSDP	Multicast Source Discovery Protocol
NAM	Network Animator
NS2	Network Simulator Version 2
NTP	Network Time Protocol
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OTcl	Object Tool Command Language
PIM	Protocol Independent Multicast
PIM-SM	Protocol Independent Multicast-Sparse Mode
PRIV	Private Extensions SDES Item
QoS	Quality of Service
RFC	Request For Comment
RIP	Routing Information Protocol
RP	Rendez-vous Point
RR	Receiver Report
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol

SDES	Source Description RTCP Packet
SDP	Session Description Protocol
SR	Sender Report
SRM	Supplier Relationship Management
SRTP	Secure Real-time Transport Protocol
SSM	Specific Source Multicast
SSRC	Synchronisation Source
Tcl	Tool Command Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
WAN	Wide Area Network

OBSAH

1 Úvod

Vzhledem k rychle se rozvíjejícímu internetu se neustále rozšiřuje počet služeb, mezi které patří také multimediální přenosy. Patří sem např. dnes hodně populární videokonferenční systémy, IP telefonie, IPTV a to jak pro jednotlivé uživatele, tak pro velké skupiny. Pro přenos dat po multimediální síti je využíván především multicastový model s RTP protokolem. Jelikož se neustále zvyšují nároky na využívané přenosové pásmo díky rozšiřujícím se informačním technologiím a především internetu je zapotřebí stále hledat a zdokonalovat nové způsoby, jak po síti přenést co největší objemy dat při co nejmenším zatížení.

Úkolem této práce je vytvořit simulační IPTV model, právě s využitím multicastu a RTP protokolu. K návrhu bude použito simulačního prostředí NS2, jehož výhodou je především to, že je volně dostupné. Je nutné prozkoumat možnosti RTP protokolu a multicastového přenosu dat v NS2. Pro tuto simulaci je zde využívána knihovna RTPSession, která používá normu rfc 1889 z roku 1996. Dalším úkolem by tedy mělo být srovnání s novou normou rfc 3550 z roku 2003, která se již běžně používá v praxi a tuto se pokusit implementovat do prostředí NS2.

2 Multicast

V dnešní době jsou sítě navrženy pro spolehlivý přenos dat, například souborů, z jednoho počítače do jiného. Požadavky na sítě jsou stále vyšší což platí i pro přenos multimediálních dat. Pro data, jako je zvuk nebo obraz nelze tolerovat větší zpoždění doručení. U sítí určených především pro přesun souborů z jednoho místa do jiného, mohou být datové pakety přenášeny nerovnoměrnou rychlostí, protože pokud části souboru přicházejí pomalu nebo v nesprávném pořadí, není to žádný větší problém. Proti tomu multimediální vyžadují, aby datové pakety přicházely do klientského počítače včas a ve správném pořadí. Tyto potíže řeší v síti protokoly pro přenos v reálném čase a služba QoS (Quality of Service). Práce s multimédii také vyžaduje přenos velkých objemů dat po síti a využívá tím větší šířku pásma sítě než základní síťové operace typu přenosu souborů. Tyto potíže řeší vysílání multicast.

2.1 Historie

Implementace multicastu se v internetu poprvé objevuje v osmdesátých letech minulého století. Steve Deering tehdy začal pracovat na distribuovaném operačním systému Vsystem. Systém se původně skládal z několika počítačů propojených mezi sebou na jednom ethernetovém segmentu. Počítače si zprávy mezi sebou posílali pomocí ethernetového multicastu. Postupným rozšiřováním sítě bylo potřeba čím dál více počítačů. Z důvodu nedostatku počítačů bylo třeba propojit více budov. Propojení s tamní sítí bylo realizované pomocí protokolu IP přes routery a tedy přímé posílání zpráv přes ethernet nebylo dále možné.

Komunikace mezi počítači musela být převedena z ethernetových rámců do světa IP paketů. Problém byl, jak ve světě IP protokolů použít skupinové adresování, multicast. Dále navrhnul způsob, jak multicastové pakety adresovat, posílat je sítí, rozšíření link-state routovacího protokolu OSPF o podporu multicastingu (MOSPF), nový vektorový routovací protokol na bázi RIP (DVMRP) a také základy protokolu IGMP.

2.2 Funkce

Přestože je multicastový paket zcela stejný jako běžný unicastový, tak ve zpracování takového paketu routerem je jedna velmi výrazná odlišnost. U unicastového paketu se router pouze jednoduše rozhodne, kterým směrem ho (pokud vůbec) přepošle. V případě multicastového routování může být více cílových stanic a router proto musí takový paket i několikrát replikovat což přirozeně routery trochu více zatěžuje. Zároveň je nutné velmi pečlivě sledovat, zda-li se toto množení neděje nadměrně, aby protistrana nedostávala svá data zbytečně ve více kopiích a neplýtvalo se tak šířkou pásma.

Koncové stroje musí mít zaregistrované multicastové adresy, aby síť věděla, kam má paket směrovaný na danou multicastovou adresu poslat. Routery pak musí být schopny spočítat optimální cestu pro průchod a duplikaci paketů pro všechny přihlášené.

Multicasting je tedy vhodný především pro aplikace, kdy jeden zdroj posílá velkému množství klientů stejná data. To se především využívá u internetového vysílání rozhlasu či televize. Data jsou vysílána z vysílače pouze multicastové skupině a o duplikace se stará síť v místě, kde to je optimální. Výrazně se tím zmenšuje zatížení sítě a vysílacího stroje.

Mezi největší slabiny multicastu patří neschopnost spolehlivého přenosu dat. Není tedy možné přes něj přenášet protokol TCP, který patří mezi spolehlivé protokoly. Pokud totiž dojde po cestě ke ztrátě dat, tak TCP striktně vyžaduje jejich opětovné posílání. Síť by si tedy musela pamatovat pakety, které již přenesla a to by neúměrně zatěžovalo routery. Další problém by byl s přenosem potvrzení o doručení od všech příjemců.

2.3 Druhy multicastu

2.3.1 ASM (Any Source Multicast)

Pro přenos dat metodou ASM musí mít síť schopnost rozlišit kdo je zdroj. V případě, že má účastník zájem o připojení do konference, musí být síť schopna určit všechny možné potenciální zdroje a jejich data pak dopravit k danému účastníkovi. Takový mechanizmus vytváří z multicastu ASM poměrně složitý systém jehož realizace není zrovna jednoduchá. Proto je nutné pro multicastové přenosy ASM vytvořit složitou strukturu směrování.

ASM přenos je vhodný především pro on-line hry, hlasové a video konference nebo sdílení elektronických tabulí, kdy se někteří účastníci střídají jako zdroje dat.

2.3.2 SSM (Specific Source Multicast)

SSM je odvozen od ASM a lze jej použít pouze v případě, kdy je zdroj dat přesně definován. Pokud by chtěl na stejnou multicastovou adresu vysílat jiný zdroj, nebude toto vysílání přijato. Toto je na straně příjemců dat zajištěno tak, že poskytnou dva typy adres: multicastovou adresu (adresa multicastové skupiny, ze které chce příjemce přijímat data) a unicastovou adresu (adresa konkrétního zdroje). Zdrojem ze kterého lze získat unicastovou adresu může být například www stránka rádiového vysílání. Výhoda SSM je v jednoduchosti a v menší náročnosti při realizaci.

Klíčovým problémem u této technologie je zkontaktovat příjemce a vysílající. V současné době existují dva způsoby, jak toho dosáhnout.

První metoda se označuje jako *dense mode* a pro názornost bychom si ji mohli pojmenovat jako paketový spam. Jakmile někdo začne vysílat na multicastovou adresu, je tímto tokem zaplavena celá síť bez ohledu na to, zda vysílání někoho zajímá nebo ne. Jednotlivé směrovače sice mají možnost požádat aby jim data, která nikoho momentálně nezajímají přestala být posílána, ale tato žádost má pouze krátkodobou účinnost a je třeba jí v případě nezájmu pravidelně opakovat. Použití takovéto metody (nejen) v Internetu je asi těžko představitelné

Druhá metoda, *sparse mode* je přijatelnější. Pro každou multicastovou skupinu je v síti ustanoveno místo, kde má příjemce možnost potkat se s vysílajícím. Toto místo je nazýváno rendez-vous pointem (RP) a je realizováno na nějakém směrovači v síti. Vysílající (přesněji jeho nejbližší směrovač) tedy posílá pakety na RP a každý zájemce o příjem dané multicastové skupiny si (prostřednictvím svého nejbližšího směrovače) pošle žádost o zasílání paketů z RP. Pakety putují tedy pouze tam, kde je o ně zájem. Navíc s prvním přijatým paketem zjistí směrovač příjemce odkud se vysílá a může si zažádat o příjem optimální cestou přímo od zdroje a zrušit příjem přes RP. Rendez-vous point v takovém případě může sloužit opravdu pouze k seznámení, tedy k tomu, aby se příjemcův směrovač dozvěděl, odkud se vysílá.

V sparse modu dokáže pracovat multicastový protokol PIM (PIM-SM) a zdálo by se, že takovéto řešení je použitelné i v prostředí Internetu. Avšak pokud chce poskytovatel zajistit svým zákazníkům kvalitní služby, nemůže se spoléhat na něco, co nemá plně ve své správě. Každý poskytovatel si pochopitelně bude chtít vytvořit RP v rámci svého autonomního systému. Vzniká tak nepříjemnost. Budou-li v Internetu stovky různých RP pro každou multicastovou skupinu, přestanou tyto plnit svou funkci a vysílající se s příjemcem nepotkají, pokud nebudou náhodou v rámci jednoho autonomního systému, tedy u stejného poskytovatele. K zajištění výměny informací mezi jednotlivými RP o tom, jaké zdroje do které multicastové skupiny momentálně vysílají slouží protokol MSDP (Multicast Source Discovery Protocol). Je však pravda, že některé rysy tohoto protokolu mohou správce autonomních systémů odrazovat od jeho nasazení.

3 Simulační prostředí NS2

3.1 Základní vlastnosti

Simulátor NS2 je zaměřen především na výzkum síťových protokolů. Je zde možné simulovat IP protokoly jako TCP, UDP, RTP, SRM a další směrovací a komunikační protokoly a to jak u drátových tak i bezdrátových sítí. Jedná se o volně šiřitelný objektově orientovaný simulátor naprogramovaný v jazyce C++ a OTcl. NS2 je projekt, který se stále vyvíjí. Postupně se nacházejí nové chyby v návrhu, které jsou opravovány. K dispozici je i zdrojový kód simulátoru, což uživatelům umožňuje přidávat podporu pro nové komunikační protokoly a monitorovací nástroje. Samotná instalace NS2 není zrovna jednoduchá, ale po instalaci můžeme začít vytvářet simulační skripty. Ty jsou psány v jazyce Tcl a kompletně popisují simulační úlohu. To znamená, že ve skriptu je kompletní popis síťové topologie, použitých komunikačních protokolů a veškerých událostí. K vytvoření takového skriptu je tedy nutná znalost vytvářeného modelu sítě, tříd simulátoru NS2 a programování v Tcl. Pokud přidáváme nový protokol do NS2, tak je potřeba kód v C++ a dále příslušný OTcl kód do datové základny simulátoru. NS2 je tedy poměrně výkonný simulátor pro analýzu celé řady protokolů na různých úrovních OSI modelu. Je vyvíjen pod licencí GNU, což považuji zároveň za výhodu i nevýhodu. Mezi nesporné výhody patří jistě to, že je volně dostupný, zdrojové kódy jsou volně k dispozici a tak se občas někomu podaří nalézt nějaké zlepšení či chybu, která je následně opravena. Nevýhodou je pak často neaktuální dokumentace právě díky postupnému vylepšování a dále poměrně vysoké paměťové nároky, kdy je každý paket během simulace uložen v operační paměti na kterou tak vznikají vysoké nároky.

3.2 Multicast v NS2

3.2.1 Centralizovaný multicast

Centralizovaný multicast je implementace neobvyklého módu multicastu podobného PIM-SM. Rendezvous Point (RP) neboli sdílený kořenový adresář, je zabudován v multicast skupině. Nastavení stavu uzlu (poslání omezení, připojující zpráva atd.) zde není simulováno.

Centralizovaný agent se používá k výpočtu přeposílání stromových struktur a k nastavení přeposílání stavu S, G jako nově obdržená připojená skupina. Datové pakety od odesílatele ke skupině jsou unicastové do RP, i když zde není příjemce pro skupinu.

Metoda zpřístupnění směrování centralizovaného multlicastu pro simulaci je následující:

```
set mproto CtrMcast          # nastaví multicast protokol
set mrthandle [$ns mrtproto $mproto]
```

Příkaz []mrtproto vytváří řízení objektu multicastového protokolu. Toto řízení může být použito pro kontrolu RP a boot-strap-router (BSR), přepínání mezi typy stromových struktur pro určitou skupinu ze sdíleného stromu k zdrojovému specifickému stromu a přepočítávání multicastové cesty.

```
$mrthandle set_c_rp $node0 $node1              # nastaví RPs
$mrthandle set_c_bsr $node0:0 $node1:1         # nastaví BSR, priority
$mrthandle get_c_rp $node0 $group              # vezme aktuální RP
$mrthandle get_c_bsr $node0                    # vezme aktuální BSR
$mrthandle switch-tree type $group             # specifikuje cestu sdíleným stromem
$mrthandle compute-mroutes                     # přepočítává cesty, obvykle vyvolaná
automaticky dle potřeby
```

Kdykoliv změna sítě vyvolá změnu unicastových cest, může být vyvolána procedura []compute-mroutes k přepočítání multicastových cest. Okamžitý přepočet představující centralizované algoritmy, může mít za následek příčinu porušení během průběhu period.

3.2.2 Dense mode

Dense Mode protokol (DM.tcl) je implementace hustého módu jako protokolu. V závislosti na hodnotě DM proměnná CacheMissMode může běžet v jednom ze dvou režimů. Pokud je CacheMissMode nastavený na pimdm (standard), budou použity pravidla přeposílání PIM-DM. Obdobně pokud je CacheMissMode nastavený na dvmrp, budou použity pravidla přeposílání DVMRP. Hlavní rozdíl mezi těmito dvěma režimy je, že DVMRP udržuje rodičovské vztahy mezi uzly, aby snížil počet spojení, přes která jsou vysílána data broadcastem. Implementace funguje na spojení point-to-point, stejně jako

LAN sítě a je přizpůsobena změnám sítě (spojení přicházející nahoru a dolů). Jakýkoliv uzel, který obdrží data z jednotlivé skupiny, pro který nemá downstream příjemcům, pošle omezený upstream. Omezená, zmenšená zpráva má za následek upstream k uzlu a zahájení omezeného stavu na tomto uzlu. Omezený stav zabraňuje tomu, že uzel posílá data pro skupinu k uzlu, který posílá originální omezenou zprávu zatímco stav je aktivní. Trvání délky po který je omezený stav aktivní je konfigurován přes DM proměnnou třídu PruneTimeout. Typická DM konfigurace je ukázána zde:

```
DM set PruneTimeout 0.3          # standardně 0.5 (sec)
DM set CacheMissMode dvmrp       # standardně pimdm
$ns mrtproto DM
```

3.2.3 Shared Tree Mode

Zjednodušený mód ST.tcl je verzí sdíleného-stromového multicast protokolu. Třída proměnných pole RP indexována skupinou určuje, který uzel je RP pro jednotlivou skupinu.

```
př.:
ST set RP_($group) $node0
$ns mrtproto ST
```

V okamžiku kdy je multicast simulace započata, protokol vytvoří a nainstaluje zapouzdřený objekt v uzlu, který má multicastové odesílatele, odpouzdří objekt v RP a spojí je. K připojení skupiny uzel pošle graft message (zpráva o připojení) směrem k RP skupiny. K opuštění skupiny pošle omezující zprávu. Protokol aktuálně nepodporuje dynamické změny v LAN.

3.2.4 Bi-directional Shared Tree Mode

BST.tcl je experimentální verze obousměrného sdíleného stromového protokolu. Stejně jako v sdíleném stromovém módu musí být RP konfigurovány ručně použitím třídního pole RP. Protokol aktuálně nepodporuje dynamické změny v LAN.

4 Změny v RFC 3550 oproti RFC 1889

Norma RFC 3550 je téměř identická s RFC 1889. Nejsou zde žádné změny ve formátech paketu, pouze změny pravidel a algoritmů řízení v užívání protokolu. Největší změnou je zlepšení dostupného časovacího algoritmu pro výpočet, kdy poslat RTCP pakety:

- Algoritmus pro výpočet RTCP přenosového intervalu uvedený dále, byl rozšířen proto, aby zahrnul "přehodnocení" minimalizovat původně zamýšlený přenos ve chvíli, kdy se do sekce připojí současně příliš mnoho účastníků a "obrácené přehodnocení" pro redukci výskytu falešného účastníka při překročení časového limitu, kdy se množství účastníků rychle snižuje. Obrácené přehodnocení je možné použít ke zkrácení zpoždění před posláním RTCP SR, při přechodu z pasivního přijímače k aktivnímu vysílajícímu módu.

- Další změnou je specifikace nových pravidel ovládání, kdy by měl být poslán RTCP BYE paket, aby se vyhnul záplavě paketů, pokud současně opustí sekci mnoho účastníků.

- Byl odstraněn požadavek k udržení stavu pro nečinné účastníky po dobu nezbytně dlouhou, aby se rozdělila síť na typické úseky. V sekcích, kde se mnoho účastníků připojuje na krátký čas a selže poslání BYE by tento požadavek způsobil významný, příliš vysoký odhad počtu účastníků. Upravený algoritmus nahrazuje velké množství nových účastníků připojujících se současně, když se jednotlivé části spojují.

Mělo by být uvedeno, že provedení těchto rozšíření má význam, jen když je počet účastníků v konferenci velký (tisíce) a pokud se většina účastníků připojí nebo odpojí zároveň. Toto testování je za provozu sítě obtížné. Nicméně algoritmus podléhá důkladné analýze a simulace ověřuje jeho výkon. Dále byl rozšířený algoritmus navržen k tomu, aby součinnost s algoritmem v RFC 1889 byla taková, aby míra snížení nadměrné RTCP šířky pásma během kroku připojení byla úměrná zlomku účastníků,

kteří realizují rozšířený algoritmus. Součinnost dvou algoritmů byla ověřena experimentálně na skutečných sítích.

4.1 Další změny ovlivňující funkci

- Implementace mohou pouze skladovat odběr vzorků SSRC identifikátorů účastníků k tomu, aby dovolily změnu měřítka velmi velkých sekcí. Tyto algoritmy jsou blíže specifikovány v RFC 2762 .

- Bylo specifikováno, že využitá a nevyužitá šířka pásma pro odesílání RTCP mohou být nastaveny jako nezávislé parametry sekce, spíše než přesně vymezené procento šířky pásma sekce a mohou být nastaveny na nulu. Požadavek, že RTCP byl povinný pro RTP sekci používající IP multicastové vysílání byl vynechán. Nicméně pro objasnění bylo dodáno, že vynechání RTCP NENÍ DOPORUČENO.

- Části vysílajících účastníků bude přiřazena RTCP šířka pásma určená z neměnné 1/4 poměru založeného na daných RTCP parametrech využité a nevyužité šířky pásma pro odesílání RTCP. Předpoklad, že odesílatelům není přiřazena žádná šířka pásma, když tam žádní odesílatelé nejsou, byl odstraněn od doby, kdy je dočasný stav očekávaný. To také udržuje nepoužívané vysílače šířky pásma RTCP od používaných, když nejsou potřeba.

- Minimální RTCP interval může být změněn na menší hodnoty pro relace s velkou šířkou pásma a počáteční RTCP zpoždění může být vynulováno pro unicastové relace.

- Časové odpojení účastníka je založeno na nečinnosti podle čísla RTCP intervalu zprávy vypočítaného používaným přijímačem z části RTCP šířky pásma stejnoměrně pro aktivní odesílatele.

- Překladače a směšovače by měly posílat BYE pakety zdrojům a nebudou je dále přeposílat.

- Jsou zde definovaná pravidla změn pro vrstvená zakódování. V posledním z nich je poznamenané, že adresa a port určují pravidlo konfliktu s SDP specifikací RFC 2327, ale je zamýšleno, že omezení bude odstraněno v opravě RFC 2327.

- Pravidlo pro používání vstupně/výstupních párů portu pro RTP a RTCP bylo objasněno, aby poukázalo na cílové vstupně/výstupní porty. Požadavek pro použití vstupně/výstupních párů portu byl odstraněn tam, kde jsou dva vstupně/výstupní porty specifikovány explicitně. Pro unicastové RTP sekce mohou být použity odlišné vstupně/výstupní páry pro dva konce.

- Byla přidána nová kapitola vysvětlující požadavky pro ochranu proti zahlcení v aplikacích používajících RTP.

- Podmínka, že musí být vybrán nový SSRC identifikátor kdykoliv je změněna zdrojová transportní adresa říká, že může být vybrán nový SSRC identifikátor. Shodně bylo objasněno to, že realizace může vybrat k řízení pakety z nové zdrojové adresy spíše než z existující zdrojové adresy, kdy nastává SSRC kolize mezi dvěma dalšími účastníky a měly by být dělány pro aplikace jako telefonie, ve které některé zdroje jako mobilní entity mohou měnit adresu během průběhu RTP relace.

- Chybný odstavce v RFC 1889 o pseudokódu pro detekce kolize a rozlišování algoritmu byl opraven přeložením syntaxe do pseudo jazyka C a algoritmus byl upraven tak, aby odstranil omezení, že RTP a RTCP musí být poslány ze stejného zdrojového čísla portu.

- Byl objasněn popis doplňovacího mechanismu pro RTCP pakety a je dáno, že doplňování musí být aplikováno pouze na poslední paket sloučeného RTCP paketu.

- Stahování mnoha ztracených paketů bylo opraveno použitím kladné a záporné hranice.

- Specifikace "příbuzného" NTP spolehlivého značení času v RTCP SR části nyní definuje tyto časové značení, aby byly založené na většině běžných specifických systémových hodinách jako systémová doba pohotovosti provozuschopnosti, spíše než na uplynulé době relace, která by nebyla stejná pro start paralelní aplikace na stejném stroji v různých časech.

4.1.1 RTCP Interval Přenosu

RTP je navržen tak, aby se aplikace automaticky přizpůsobovala během relace od několika účastníků až po několik tisíc. Např. během audio konference je datový tok samoomezující, protože pouze jeden nebo dva lidé současně hovoří, takže rychlost přenosu dat u multicastu na jakémkoliv daném spojení zůstává relativně konstantní nezávisle na počtu účastníků. Ale řízení toku není samoomezující. Pokud byly zprávy příjemce od každého účastníka poslány konstantní rychlostí, řízení toku by rostlo lineárně s počtem účastníků. Tudíž rychlost musí být snižována dynamickým výpočtem intervalu mezi přenosy RTCP paketů.

Pro každou relaci se předpokládá, že datový tok podléhá agregovanému limitu nazývanému „šířka pásma relace“, který se dělí mezi účastníky. Tato šířka pásma může být vyhrazena a limit vynucený sítí. Pokud není šířka pásma vyhrazena mohou existovat jiná omezení závislé na prostředí, která zavedou „rozumné“ maximum použité v relaci jako šířku pásma. Šířka pásma může být vybrána podle určité ceny nebo znalosti dostupné šířky pásma sítě pro relaci. Nezávisí to tolik na kódování média, ale výběr kódování může být omezen šířkou pásma. Často je šířka pásma sumou nominální šířky pásma odesilatelů, u kterých se očekává, že budou současně aktivní. Pro audio telekonference by toto číslo typicky bylo šířkou pásma jednoho odesilatele. Pro vrstvové kódování je každá vrstva oddělená RTP relace se svým vlastním parametrem šířky pásma relace.

Předpokládá se, že parametr šířky pásma relace je opatřený aplikací managementu relace když jej vyvolá mediální aplikace, ale mediální aplikace může nastavit defaultní hodnotu založenou na šířce pásma jednotlivého odesilatele pro kódování vybrané pro relaci. Aplikace také může vynutit omezení šířky pásma založené na pravidlech multicastu nebo jiných kritériích. Všichni účastníci musí použít stejnou hodnotu pro šířku pásma relace, aby byl vypočítaný stejný RTCP interval.

Výpočet šířky pásma pro řízení toku a dat zahrnuje síťové protokoly a přenosy v nižších vrstvách (např. UDP a IP), protože toto potřebuje znát systém rezervace zdrojů. Také se předpokládá, že aplikace potřebuje vědět, které protokoly se používají. Odkazy úrovní hlaviček nejsou zahrnuty ve výpočtu, protože paket bude zapouzdřen během své cesty s různými odkazy úrovní hlaviček.

Řízení toku by mělo být omezeno na malé a známé zlomky šířky pásma relace: malé, aby nebyla narušena primární funkce transportního protokolu přenášet data; známé, aby řízení toku mohlo být zahrnuto ve specifikaci šířky pásma dané protokolu rezervace zdrojů a aby každý účastník mohl nezávisle vypočítat svůj díl. Šířka pásma řízení toku je navíc k šířce pásma relace pro datový tok. Doporučuje se, aby část šířky pásma relace přidaná pro RTCP byla fixní 5%. Také se doporučuje, aby ¼ šířky RTCP pásma byla vyhrazena účastníkům, kteří odesílají data, aby v relacích s velkým počtem příjemců, ale malým počtem odesilatelů mohli nově připojení účastníci obdržet své CNAME pro odesílací strany rychleji. Když je poměr odesilatelů větší než ¼ účastníků, obdrží odesilatelé svůj podíl plné RTCP šířky pásma. I když hodnoty těchto a jiných konstant ve výpočtu intervalu nejsou kritické, všichni účastníci v relaci musí použít stejné hodnoty, aby byly vypočteny stejné intervaly. Tudíž tyto konstanty by měly být fixní pro konkrétní profil.

Profil může specifikovat, že řízení toku šířky pásma může být samostatný parametr relace než striktní procento šířky pásma relace. Použití samostatného parametru umožňuje přizpůsobujícím se aplikacím nastavit šířku RTCP pásma konzistentní s „typickou“ šířkou pásma, která je menší než maximální šířka specifikovaná parametrem šířky pásma relace.

Profil dále může specifikovat, že šířka pásma řízení toku může být rozdělena do dvou samostatných parametrů relace pro ty účastníky, kteří jsou aktivními odesílateli dat a těmi, kteří jimi nejsou; nazývejme ty parametry S a R. Vezmeme-li v úvahu doporučení, že ¼ šířky RTCP pásma je oddělena pro odesilatele dat, pak doporučené defaultní hodnoty pro tyto dva parametry jsou 1.25% respektive 3.75%. Když je podíl odesilatelů větší než S/(S+R) účastníků, odesilatelé dostanou svůj podíl sumy těchto parametrů. Použití dvou parametrů umožňuje RTCP příjemci zpráv úplné odpojení v určité relaci nastavením RTCP šířky pásma pro ne-datové-odesilatele na nulu zatímco pro odesilatele dat se udržuje ne-nulová RTCP šířka pásma, aby zprávy odesilatele mohly být stále odeslány pro inter-média synchronizaci. Vypnutí RTCP příjmu zpráv se nedoporučuje, protože jsou potřebné zvláště pro informace o kvalitě příjmu a řízení přetížení. Nicméně toto může být vhodné pro systémy operující na jednosměrných spojích nebo pro relace, které nevyžadují zpětnou vazbu kvůli kvalitě příjmu nebo životnosti příjemců a které mají jiné prostředky pro zabránění přetížení.

Vypočítaný interval mezi přenosy složených RTCP paketů by měl také mít spodní meze, aby se zabránilo explozi paketů a překročení šířky pásma když je počet účastníků malý a provoz není plynulý podle zákona velkých čísel. Také se udržuje interval zpráv, aby nebyl příliš malý během přechodných výpadků jako síťová partition, kdy je adaptace pozdržená během zotavení partition. Při startu aplikace by se mělo zavést zpoždění, než je první složený RTCP paket odeslán, aby se získal čas pro přijetí RTCP paketů od dalších účastníků, takže interval zpráv se přiblíží správné hodnotě rychleji. Toto zpoždění může být nastaveno na polovinu minimálního intervalu, aby se dosáhlo rychlejší notifikace, že je přítomen nový účastník. Doporučená hodnota pro fixní minimální interval je 5 sekund.

Implementace může měnit velikost minimálního RTCP intervalu na menší hodnotu v inverzním poměru k šířce pásma relace s následujícími omezeními:

- Pro multicastové relace pouze aktivní odesilatelé dat mohou použít redukovanou minimální hodnotu k výpočtu intervalu pro přenos složených RTCP paketů.

- Pro unicastové relace mohou redukovanou hodnotu použít pouze účastníci, kteří rovněž nejsou aktivními odesilateli dat a zpoždění před odesláním počátečního složeného RTCP paketu může být nulové.

- Pro všechny relace by se mělo použít fixní minimum při výpočtu účastníkova timeout intervalu, aby implementace, které nepoužívají redukovanou hodnotu pro přenos RTCP paketů nebyly předčasně odpojeny dalšími účastníky.

- Doporučená hodnota pro redukované minimum v sekundách je 360 děleno šířkou pásma relace v kilobytech/s. Toto minimum je menší než 5 sekund pro šířky pásma větší než 72 kb/s.

Algoritmus popsaný v pravidlech odeslání a přijímání paketu byl navržen pro záměry nastíněné v této kapitole. Vypočítává interval mezi odesláním složených RTCP paketů, aby se povolená šířka pásma rozdělila mezi účastníky. To dovoluje aplikaci poskytnout rychlou odpověď pro malé relace kde např. je důležitá identifikace všech účastníků a přitom se automaticky přizpůsobuje velkým relacím. Algoritmus obsahuje následující charakteristiky:

- Vypočítaný interval mezi RTCP pakety mění velikost lineárně s počtem členů ve skupině. Je to tento lineární faktor, který umožňuje konstantní velikost řízení provozu v souhrnu všech členů.

- Interval mezi RTCP pakety se náhodně mění v rozsahu [0.5,1.5] krát vypočítaný interval, aby se zabránilo nechtěné synchronizaci všech účastníků. První odeslaný RTCP paket po připojení se k relaci je také zpožděn náhodnou proměnnou polovinou minimálního RTCP intervalu.

- Dynamický odhad průměrné velikosti složeného RTCP paketu je vypočítaný včetně všech přijatých a odeslaných paketů, aby se automaticky přizpůsobil změnám v množství přenášených řídících informací.

- Protože je vypočítaný interval závislý na počtu sledovaných členů skupiny, mohou se vyskytovat nežádoucí inicializační efekty, když se nový uživatel připojí do existující relace, nebo když se mnoho uživatelů současně připojí do nové relace. Tito noví uživatelé budou mít na počátku nesprávný odhad členství ve skupině a tudíž jejich RTCP přenosový interval bude příliš krátký. Tento problém může být závažný pokud se mnoho uživatelů připojí současně k relaci. K vyřešení této záležitosti je zaveden algoritmus nazývaný „timer reconsideration“. Tento algoritmus implementuje jednoduchý back-off mechanismus, který potlačuje přenos RTCP paketů uživatelů pokud se velikost skupiny zvětšuje.

- Když uživatelé opustí relaci buď pomocí BYE nebo timeout, počet členů ve skupině se snižuje a i vypočítaný interval by se měl snižovat. Je použit „reverse reconsideration „ algoritmus, který dovoluje členům rychleji redukovat své intervaly v odpovědi na snižování členů ve skupině.

- S BYE pakety se zachází jinak než s ostatními RTCP pakety. Když uživatel opustí skupinu a přeje si odeslat BYE paket může to učinit před dalším plánovaným RTCP paketem. Nicméně, přenos BYE paketů se řídí podle back-off algoritmu, který zabraňuje záplavě BYE paketů v případě, že velký počet členů současně opouští relaci.

Tento algoritmus může být použit pro relace, ve kterých mohou všichni účastníci odesílat. V tomto případě je parametr šířky pásma relace produktem šířky pásma individuálního odesilatele krát počet účastníků a šířka RTCP pásma je z toho 5%.

Udržování počtu členů relace

Výpočet intervalu RTCP paketu závisí na odhadu počtu stran účastnících se relace. Nové strany jsou připočítány, když jsou zjištěny a zápis pro každou z nich by měl být vytvořen v tabulce indexované SSRC a CSRC identifikátorem kvůli sledovatelnosti. Nové zápisy nemusí být považovány za validní do doby než jsou přijaty vícenásobné pakety nesoucí nové SSRC, nebo než je přijat SDES RTCP paket obsahující CNAME pro dané SSRC. Zápisy mohou být z tabulky vymazány při přijetí RTCP BYE

paketu s odpovídajícím SSRC identifikátorem, s výjimkou zatoulaných paketů, které dorazí po BYE a zapříčiní obnovení zápisu. Kromě toho zápis by měl být označený příjmem BYE a po příslušném zpoždění smazán.

Účastník může označit jinou stranu neaktivní nebo smazat pokud ještě není validní jestliže nebyl přijat žádný RTP nebo RTCP paket pro malý počet RTCP intervalů zpráv (5 je doporučeno). To poskytuje určitou odolnost proti ztrátě paketů. Všechny strany musí mít pro tento multiplikátor stejnou hodnotu a musí počítat zhruba stejnou hodnotu pro RTCP interval zpráv, aby timeout správně pracoval. Tudíž tento multiplikátor by měl být fixní pro určitý profil.

Pro relace s velkým počtem účastníků může být nepraktické udržovat tabulku k ukládání SSRC identifikátoru a stavové informace pro všechny z nich. Implementace může použít SSRC vzorkování ke snížení požadavků na ukládání. Implementace může použít jakýkoliv jiný algoritmus s obdobným chováním. Klíčovým požadavkem je, aby jakýkoliv uvažovaný algoritmus výrazně nepodhodnotil velikost skupiny ačkoliv ji může nadhodnotit.

4.1.2 RTCP pravidla odeslání a přijetí paketu

Zde jsou nastíněna pravidla jak odeslat a co dělat při příjmu RTCP paketu. Implementace, která umožňuje operace v multicastovém prostředí nebo v prostředí multipoint unicast musí splňovat požadavky uvedené výše. Taková implementace může použít algoritmus definovaný v této sekci pro splnění těchto požadavků nebo může použít jiný algoritmus pokud poskytne ekvivalentní nebo lepší výkon. Implementace, která je omezená na dvoustranou unicast operaci by přesto měla použít náhodnost RTCP přenosového intervalu, aby se zamezilo nechtěné synchronizaci vícenásobných instancí operujících ve stejném prostředí, ale může vynechat algoritmy „timer reconsideration“ a „reverse reconsideration“.

K vykonání těchto pravidel musí účastník relace udržovat několik druhů stavu:

- tp: poslední čas přenosu RTCP paketu,
- tc: aktuální čas,

- tn: další plánovaný přenosový čas RTCP paketu,
- pmembers: odhadovaný počet členů relace v čase posledního výpočtu tn,
- members: nejaktuálnější odhad počtu členů relace,
- senders: nejaktuálnější odhad počtu senders relace,
- rtcp_bw: cílová šířka RTCP pásma, tj. celková šířka pásma, která bude použita pro RTCP pakety všemi členy této relace v oktetech za sekundu. Toto bude specifikovaný zlomek parametru „šířka pásma relace“ dodaného aplikací při startu,
- we_sent: Flag který se rovná true, jestliže aplikace poslala data od doby, kdy byla přenesena 2 předcházející RTCP zpráva,
- avg_rtcp_size: průměrná velikost složeného RTCP paketu v oktetech ze všech RTCP paketů poslaných a přijatých tímto účastníkem. Velikost zahrnuje hlavičky přenosových a síťových protokolů nižší vrstvy (např. UDP a IP) jak bylo vysvětleno výše.
- initial: Flag rovná se true pokud aplikace ještě neposlala RTCP paket.

Mnohé z těchto pravidel používají „vypočtený interval“ mezi přenosy paketů. Tento interval je popsán v následující části.

Výpočet RTCP Přenosového Intervalu

Pro udržení rozšiřitelnosti by měl průměrný interval mezi pakety od účastníka relace měnit velikost podle velikosti skupiny. Tento interval se nazývá vypočítaný interval. Získává se kombinací počtu kusů stavu popsaných výše. Vypočítaný interval T je tedy stanoven takto:

1. Pokud je počet odesilatelů menší než nebo roven 25% členů, závisí interval na tom zda je či není odesilatel účastníkem (podle hodnoty we_sent). Pokud je účastník odesilatelem (poslali jsme true), konstanta C je nastavena na průměrnou velikost RTCP paketu (avg_rtcp_size) dělenou 25% RTCP šířky pásma (rtcp_bw) a konstanta n je nastavena na počet odesilatelů. Pokud we_sent není true, konstanta C je nastavena na průměrnou velikost RTCP paketu dělenou 75% RTCP šířky pásma. Konstanta n je nastavena na počet příjemců (members – senders). Pokud je počet odesilatelů větší než 25%, jsou

odesilatelé a příjemci zpracováváni společně. Konstanta C je nastavena na průměrnou velikost RTCP paketu dělenou celkovou RTCP šířkou pásma a n je nastavena na celkový počet členů. Jak je poznamenáno výše, RTP profil může specifikovat, že RTCP šířka pásma může být explicitně definována dvěmi samostatnými parametry (nazývají se S a R) pro ty účastníky, kteří jsou i nejsou odesilateli. V tom případě je 25% podíl S/(S+R) a 75% podíl je R/(S+R). Povšimněte si, že pokud je R nula, tak procento odesilatelů není nikdy větší než S/(S+R) a implementace se musí vyhnout dělení nulou.

2. Pokud účastník ještě neodeslal RTCP paket (proměnná initial je true), je konstanta Tmin nastavená na 2,5 sekundy, jinak na 5 sekund.

3. Deterministický vypočítaný interval Td je nastavený max (Tmin, n*C).

4. Vypočítaný interval T je nastavený na číslo rovnoměrně distribuované mezi 0.5 a 1.5 krát deterministický vypočítaný interval.

5. Výsledná hodnota T je vydělena e-3/2=1.21828, aby se kompenzovalo to, že přehodnocovací algoritmus časování se blíží hodnotě RTCP šířky pásma, která je pod zamýšleným průměrem.

Výsledkem této procedury je interval, který je náhodný, ale který v průměru dává nejméně 25% RTCP šířky pásma odesilatelům a zbytek příjemcům. Pokud odesilatelé tvoří více než jednu čtvrtinu členství, tato procedura rozdělí šířku pásma rovnoměrně mezi všechny účastníky.

Inicializace

Při připojení se k relaci účastník inicializuje tp na 0, tc na 0, senders na 0, pmembers na 0, members na 1, we_sent na false, rtcp_bw na specifický podíl šířky pásma relace, initial na true a avg_rtcp_size na pravděpodobnou velikost prvního RTCP paketu, který aplikace později vytvoří. Vypočítaný interval T je tedy vypočítán a první paket je naplánovaný na čas tn=T. To znamená, že časovač relace je nastavený a vyprší

v čase T. Aplikace může použít jakýkoliv požadovaný přístup pro implementaci tohoto časovače.

Účastník přidá své vlastní SSRC do member tabulky.

Příjem RTP nebo Non-BYE RTCP Paketu

Když je přijatý RTP nebo RTCP paket od účastníka jehož SSRC není v member tabulce, je přidáno do tabulky a hodnota pro members je po validaci účastníka aktualizovaná, jak je popsáno výše. Tentýž postup se objevuje pro každý CSRC ve validním RTP paketu.

Když je přijatý RTP nebo RTCP paket od účastníka jehož SSRC není v tabulce odesílatele, je přidáno do tabulky a hodnota pro odesilatele je aktualizována.

Pro každý přijatý složený RTCP paket je aktualizována hodnota avg_rtcp_size:

avg_rtcp_size = (1/16) * packet_size + (15/16) * avg_rtcp_size,
kde packet_size je velikost právě přijatého RTCP paketu.

Příjem RTCP BYE Paketu

S výjimkou toho co je napsáno v kapitole 4.1.2 pro případ, když má být přenášen RTCP BYE, pokud je přijatý paket RTCP BYE paket, zkontroluje se SSRC podle members tabulky. Pokud je přítomno je záznam z tabulky odstraněn a hodnota pro members je aktualizována. Pak se SSRC zkontroluje podle tabulky odesilatele. Pokud je přítomno, záznam je z tabulky odstraněn a hodnota pro odesilatele je aktualizována.

Kromě toho, aby se přenosová rychlost RTCP paketů více přizpůsobovala změnám ve skupinovém členství, měl by se vykonat následující „opačně přehodnocovací“ algoritmus po přijetí BYE paketu, který redukuje members na hodnotu menší než pmembers:

- Hodnota pro tn je aktualizována podle následujícího vzorce:

  tn = tc + (members/pmembers) * (tn - tc)

- Hodnota pro tp je aktualizována podle následujícího vzorce:
  tp = tc - (members/pmembers) * (tc - tp).

- Další RTCP paket je přeplánován pro přenos v čase tn, který je nyní časnější.

- Hodnota pmembers je nastavena rovnající se members.

Tento algoritmus nezabraňuje tomu, aby odhad velikosti skupiny nesprávně klesl na nulu po krátkou dobu kvůli předčasnému timeoutu, když většina účastníků velké relace ji najednou opustí, ale někteří zůstanou. Algoritmus způsobuje rychlejší návrat odhadu ke správné hodnotě. Tato situace je dost neobvyklá a následky jsou dostatečně neškodné, takže je tento problém považován za druhořadý.

Timing Out SSRC

Účastníci musí v občasných intervalech kontrolovat, jestli některý z dalších účastníků není odpojen. Účastník vypočítá deterministický (bez náhodného faktoru) počítaný interval Td pro příjemce, tj. we_sent rovná se false. Všichni ostatní účastníci, kteří neposlali RTP nebo RTCP paket od času tc – MTd (M je timeout násobitel a defaultní hodnota je 5) jsou odpojeni. To znamená, že jejich SSRC je odstraněno z member seznamu a members je aktualizováno. Obdobná kontrola se provádí u seznamu odesilatele. Kterýkoliv member na seznamu odesilatele, který neposlal RTP paket od času tc – 2T (během posledních dvou RTCP intervalů zpráv) je odstraněn ze seznamu odesilatele a odesílání je aktualizováno.

Pokud kterýkoliv member je ve stavu time out, pak se provede opačně přehodnocovací algoritmus popsaný v výše.

Účastníci musí provést tuto kontrolu nejméně jednou za RTCP přenosový interval.

Expirace časovače přenosu

Když paket časovače přenosu vyprší, účastník provede následující operaci:

- Přenosový interval T je vypočítán dle popisu výpočtu RTCP přenosového intervalu včetně randomizačního faktoru.

- Pokud je tp + T menší nebo roven tc, je RTCP paket přenesen. tp je nastaveno na tc, pak se vypočítá další hodnota T jako v předchozím kroku a tn je nastaveno na tc + T. Časovač přenosu je nastaven tak, aby vypršel opět v čase tn. Pokud tp + T je větší než tc, tn je nastaveno na tp + T. Není přenášen žádný RTCP paket. Časovač přenosu je nastaven tak, aby vypršel v čase tn.

- pmembers je nastaveno na members.

Pokud je přenášen RTCP paket, je hodnota initial nastavena na FALSE. Kromě toho je aktualizována hodnota of avg_rtcp_size:

avg_rtcp_size = (1/16) * packet_size + (15/16) * avg_rtcp_size,

kde packet_size je velikost RTCP paketu, který je právě přenášen.

Přenos BYE Paketu

Když si účastník přeje opustit relaci, přenáší se BYE paket, který informuje ostatní účastníky o této události. Aby se předešlo záplavě BYE paketů, když mnoho účastníků opouští systém, musí účastník provést následující algoritmus, pokud je počet účastníků větší než 50 v okamžiku, kdy účastník odchází. Tento algoritmus si přivlastňuje normální roli proměnné members k počítání BYE paketů:

- Když se účastník rozhodne opustit systém, tp je resetováno na tc, current time, members a pmembers jsou nastaveny na 1, initial se nastaví na 1, we_sent se nastaví na false, odesílání se nastaví na 0 a avg_rtcp_size se nastaví na velikost složeného BYE paketu. Spočítá se vypočítaný interval T. BYE paket je pak načasován na čas tn = tc + T.

- Pokaždé když je přijat BYE paket od jiného účastníka, members se zvyšuje o 1 bez ohledu na to, zda účastník existuje v members tabulce, a když se používá SSRC

vzorkování, bez ohledu na to, zda BYE SSRC byl obsažen ve vzorku. Members se nezvyšuje, když se přijímají další RTCP nebo RTP pakety, ale pouze pro BYE pakety. Obdobně avg_rtcp_size se aktualizuje pouze pro přijaté BYE pakety. Odesílání se neaktualizuje při příchodu RTP paketů, zůstává na 0.

- Přenos BYE paketů pak dodržuje pravidla pro přenášení regulérních RTCP paketů, viz výše.

To umožňuje BYE paketům, aby byly ihned odeslány, ale přesto se řídí jejich užití celkové šířky pásma. V nejhorším případě to může způsobit, že RTCP řídící pakety použijí šířku pásma dvakrát větší než je normální (10%) -- 5% pro non-BYE RTCP pakety a 5% pro BYE.

Účastník, který nechce čekat na výše uvedený mechanismus k umožnění přenosu BYE paketů, může opustit skupinu aniž vůbec BYE paket odešle. Tento účastník bude nakonec odpojen jinými členy skupiny.

Pokud je odhadovaná velikost skupiny members menší než 50, když se účastník rozhodne odejít, může účastník odeslat BYE paket okamžitě. Alternativně si účastník může zvolit vykonání výše uvedeného BYE back-off algoritmu.

V obou případech účastník, který nikdy neposlal RTP nebo RTCP paket nesmí poslat BYE paket, když opouští skupinu.

Aktualizace we_sent

Proměnná we_sent je true pokud účastník poslední dobou odeslal RTP paket, jinak je false. Toto vymezení je dáno použitím stejného mechanismu jako pro management souboru ostatních účastníků zapsaných v senders tabulce. Pokud účastník odešle RTP paket kde we_sent je false přidá se do sender tabulky a nastaví we_sent na true. Algoritmus reverse reconsideration popsaný výše by měl být vykonán, aby eventuelně snížil zpoždění před odesláním SR paketu. Pokaždé, když je odeslán další RTP paket, čas přenosu tohoto paketu je udržován v tabulce. Normální sender timeout algoritmus je pak aplikován na účastníka – jestliže RTP paket nebyl přenesen od času tc -

2T, účastník se odstraní ze sender tabulky, dekrementuje sender count a nastaví we_sent na false.

Přidělení Source Description Bandwidth

Tato specifikace definuje několik source description (SDES) položek kromě povinného CNAME jako je NAME (osobní jméno) a EMAIL (e-mailová adresa). Také poskytuje prostředky pro definování nových pro aplikaci specifických RTCP typů paketů. Aplikace by měly obezřetně přidělovat šířku pásma k této dodatečné informaci, protože to zpomalí rychlost, kterou se odesílají zprávy a CNAME a tím se zhorší výkon protokolu. Doporučuje se nepoužívat více než 20% přidělené RTCP šířky pásma jednomu účastníkovi pro přenos této dodatečné informace. Dále, není záměrem, aby všechny SDES položky byly obsaženy v každé aplikaci. Těm, které jsou obsaženy by měl být přidělen zlomek šířky pásma podle jejich užitečnosti. Raději než odhadovat tyto zlomky dynamicky, doporučuje se, aby procenta byly staticky přeložena do počtu intervalu zprávy založené na typické délce položky.

Například, aplikace může být navržená pro odesílání pouze CNAME, NAME a EMAIL. NAME může být přidělena mnohem vyšší priorita než EMAIL, protože NAME se zobrazuje nepřetržitě v uživatelově aplikačním rozhraní zatímco EMAIL je zobrazen pouze na vyžádání. V každém RTCP intervalu se odesílá RR paket a SDSE paket spolu s CNAME. U malých relací operujících s minimálním intervalem to je v průměru každých 5 sekund. Každý třetí interval (15 sekund) se v SDES paketu objevuje jedna extra položka. V sedmi z devíti případů to je položka NAME a každý devátý případ (2 minuty) to je položka EMAIL.

Když vícenásobné aplikace operují ve shodě s používáním cross-application svázanými skrz běžné CNAME pro každého účastníka, např. v multimediální konferenci skládající se z RTP relace pro každé médium, dodatečná SDES informace může být poslána pouze v jedné RTP relaci. Ostatní relace ponesou pouze položku CNAME. Tento přístup by měl být aplikován ve vícenásobných relacích layered encoding schneme.

Zprávy odesilatele a příjemce

RTP příjemci poskytují zpětnou vazbu o kvalitě pomocí RTCP paketů zpráv, které mohou mít jednu nebo dvě formy podle toho jestli je příjemce také odesílatel. Jediný rozdíl mezi SR a RR kromě typu kódu paketu je, že zpráva odesilatele obsahuje 20-ti bytovou sender information sekci pro použití aktivními senders. SR se vystavuje jestliže strana odeslala jakékoli datové pakety v intervalu od vystavení posledního nebo předcházejícího reportu jinak se vystavuje RR.

Jak SR tak RR obsahují žádnou nebo více bloků zpráv příjemce, jednu pro každý synchronizační zdroj, ze kterého tento příjemce přijal RTP datové pakety od poslední zprávy. Zprávy nejsou vystaveny pro přispívající zdroje uvedené v CSRC seznamu. Každý blok příjmu zpráv poskytuje statistiky o přijatých datech z určitého zdroje uvedeného v tomto bloku. Protože se do jednoho SR nebo RR paketu vejde maximálně 31 bloků příjmu zpráv, dodatečné RR pakety by měly být stohovány po počátečním SR nebo RR paketu podle potřeby obsahovat zprávy příjemce pro všechny zdroje slyšené během intervalu od poslední zprávy. Pokud je příliš mnoho zdrojů, aby se vešly do všech potřebných RR paketů n jednom složeném RTCP paketu aniž by přesáhly MTU síťové cesty, pak by v každém intervalu měl být pouze ten subset, který se vejde do jednoho MTU. Subsety by měly být vybrány pomocí round-robin vícenásobných intervalů, aby se reportovaly všechny zdroje.

Další sekce definuje formáty těchto dvou reportů, jak mohou být rozšířeny specifickým způsobem, jestliže aplikace vyžaduje dodatečnou zpětnovazební informaci a jak mohou být použity reporty.

4.2 Změny, které nemají vliv na funkci

- Přijímač musí ignorovat pakety s typem užitečného zatížení, kterým nerozumí.

- Byla opravena pohyblivá desetinná čárka NTP hodnoty časové značky, nějaké chybějící počáteční nuly byly přidány v hexadecimálním čísle a bylo specifikováno UTC časové pásmo.

- Je vysvětlena nedůslednost NTP časových značek přenášeného textu v roce 2036.

- Zásada pro registraci RTCP typu paketu a typu SDES byla objasněna v nové kapitole. Návrh, že výzkumní pracovníci registrují čísla, která potřebují a neregistrují ty prokazatelně nepotřebné, byl odstraněn ve prospěch používání APP a PRIV. Také byla specifikována registrace jmen profilů.

- Doporučení pro znakovou sadu UTF-8 byla změněna z X/Open předběžné specifikace na RFC 2279.

- Doporučení pro RFC 1597 bylo aktualizováno v RFC 1918 a doporučení pro RFC 2543 bylo aktualizováno v RFC 3261.

- Poslední odstavec úvodu v RFC 1889, který varuje vývojáře v omezení rozmístění v Internetu byl odstraněn, protože již nebyl považován za důležitý.

- Byla přidána nová poznámka netýkající se normy s ohledem na použití RTP se Source Specific Multicastem (SSM).

- Definice "RTP relace" byla rozšířena o tvrzení, že jednotlivá relace může užívat vícenásobnou cílovou transportní adresu (což byl vždy případ pro překladač nebo směšovač) a o vysvětlení, že rozlišování uvedené RTP relace odpovídá separátnímu SSRC identifikátoru prostoru. Byla přidaná nová definice "multimediální relace", aby redukovala nesrovnalosti o slovu "relace".

- Význam "rychlého vzorkování" byl vysvětlen podrobněji v části definice pole časové značky RTP.

- Na několika místech v textu byly objasněny otázky čtenářů. Zvláště pak tyto:

- V RFC 1889 bylo ztraceno prvních pět slov druhé věty kapitoly 2.2 při zpracování dokumentu ze zdroje výstupního formátu, ale nyní jsou obnoveny.

- V kapitole 3 byla přidána definice "RTP mediálního typu" pro vysvětlení multiplexování RTP relace v kapitole 5.2, která se týká objasnění multiplexování vícenásobných prostředků. Tato kapitola rovněž vysvětluje, že multiplexování vícenásobných zdrojů stejného prostředku založeného na SSRC identifikátorech smí být přiměřené a je normováno pro multicastové relace.

- Popis parametru relace šířky pásma je rozšířen v kapitole 6.2, včetně objasnění, že kontrola provozu šířky pásma je přičtena k relaci šířce pásma pro přenášená data.

- Byl vysvětlen efekt proměnné životnosti paketu na výpočet kolísání.

- Byla objasněna metoda pro ukončování a vyplňování sekvence SDES položek.

- Byly přidány příklady adres IPv6 v popisu SDES CNAME a "example.com" byl použit na místě dalších příkladů doménových jmen.

- Bezpečnostní část přidala formální doporučení na IPsec, která je nyní k dispozici a říká, že metoda důvěrnosti definovaná v této specifikaci je v první řadě k systematickému třídění současných postupů. Je doporučeno, aby byly užívány silnější šifrovací algoritmy jako Triple–DES, místo standardního algoritmu a poznamenáno, že SRTP profil založený na AES bude do budoucna správná volba. Bylo přidáno upozornění na slabost RTP hlavičky jako inicializačního vektoru. Bylo poznamenáno, že je nezbytné počítat šifrování pouze užitečného zatížení pro kompresi hlavičky.

- Byla objasněna metoda pro částečné šifrování RTCP; v jednotlivých částech SDES CNAME je přenášen jen v jedné části, kdy je sestavený RTCP paket rozdělený.

- Je jasné, že by mělo být posláno pouze jedno sloučení RTCP paketu pomocí reportujícího intervalu, a že je–li příliš mnoho aktivních zdrojů pro zprávy k vyhodnocení MTU, pak by měla být vybrána cyklická obsluha podskupiny zdrojů přes vícenásobné intervaly.

- Byla přidána poznámka, že pakety mohou být uloženy během ověřování platnosti RTP hlavičky a úspěšně doručeny.

- Směšovač hromadících se SDES paketů užívá větší RTCP šířku pásma kvůli delším paketům a směšovač procházející RTCP přirozeně posílá pakety ve vyšším než jednotlivém poměru zdrojů, ale platná jsou obojí chování.

- RTP aplikace může používat vícenásobné profily, ale typicky jen v jedné dané relaci.

- Výrazy MUST, SHOULD, MAY, atd. jsou užívány podle definic v RFC 2119.

5 Návrh

Pro návrh a následnou simulaci jednoduché sítě byl vybrán simulátor NS2. Pro své přednosti je zde využit multicast CtrMcast, který je implementován v NS2 a pro zajištění přenosu dat bude použit RTP transportní protokol. NS2 v současnosti používá starší normu rfc 1889. V dnešní době se však již běžně používá v praxi novější norma rfc 3550.

5.1 Transportní protokoly – RTP, RTCP

Úkolem transportního protokolu je přenést multimediální data od zdroje k cíli. Přenos multimédií využívá protokol RTP (Real Time Protocol – RFC 1889). Před vytvořením paketu je signál digitalizován (pokud je to nutné) a zpracován některým ze standardních algoritmů. Běžně jsou používány G.711, G.726, GSM-EFR, G.729, G.723. To jaký algoritmus zvolíme má vliv na šířku pásma a kvalitu přeneseného signálu. Čím nižší šířka pásma, tím náročnější zpracování pro procesor. Digitalizací a kódováním je původní signál zkreslen.

RTP pracuje jako nadstavba protokolu UDP, ale navíc přidává sekvenční a synchronizační informace. UDP na rozdíl od TCP neobsahuje žádný samoopravný mechanizmus. Protokol byl zvolen záměrně, protože cílem transportu multimédií není zajistit úplnost přenášené informace, ale její doručení v reálném čase. Proto zde není třeba spolehlivého TCP protokolu, jehož opravné mechanizmy by byly spíše překážkou přenosu.

Přesný popis polí hlavičky je popsán v RFC 1889, ty nejdůležitější ale jsou:

Sequence number – sekvenční číslo pro zjištění, zda nedošlo při transportu ke ztrátě paketů nebo ke změně jejich pořadí,

RTP timestamp – synchronizační značka, její rozlišení záleží na povaze přenášené informace a je na signalizačním protokolu, aby ho zajistil před začátkem přenosu,

Synchronization source ID – identifikátor synchronizačního zdroje pro případ, že se přenáší více kanálů najednou (např. hlas a video).

Úlohou RTCP je přenést informaci od přijímače k vysílači o tom, v jaké kvalitě je signál přijímán, tj. kolik se cestou ztratilo paketů a jaké bylo proměnné zpoždění těch, které se přenesly. RTCP nemá opravnou funkci jako TCP, ale pouze dává vysílající aplikaci informaci o kvalitě příjmu. Ta pak musí sama rozhodnout, zda např. kvůli velkým ztrátám paketů nezmění kódování signálu tak, aby byl přenesen třeba v nižší kvalitě, ale s menšími ztrátami.

Transportní hlavičky mohou představovat značnou část paketů, celkově zabírají 40 B (20 pro IP, 8 pro UDP a 12 pro RTP), oproti tomu samotná přenášená informace může zabírat pouze 20 B. Proto byla vyvinuta komprese RTP hlaviček, která vytvoří z původních 40 B 2 – 4 podle toho, jak velký je rozdíl informací v po sobě jdoucích paketech. Tuto metodu lze využít pouze u dvoubodových spojů, které jsou typické pro sítě WAN, kde ale také vzniká největší potřeba pro šetření pásmem.

Přijímací strana pak sestaví jednotlivé vzorky, dekóduje a vyšle je jako signál dále. Nejdůležitější částí je tzv. dejitter buffer, což je paměť pro vyrovnávání proměnného zpoždění, ke kterému došlo vlivem různého zpoždění jednotlivých paketů při přenosu (např. ve výstupních frontách směrovačů a přepínačů).

RTP definuje standardní paketový formát pro doručování multimediálních dat po internetu. Původně byl vyvinut jako protokol pro výběrové vysílání, ale byl používán v mnoha unicast aplikacích. Protokol se často používá v systémech s prouděním mediálních dat ve spojení s RTSP jako video telefonní konference nebo videokonference a v push to talk systémech.

Služby pracující s RTP zahrnují určení užitečného zatížení, číslování sekvencí, časové razítkování a sledování přenosu. Novější Internet Standard definovaný v RFC 3550 dělá RFC 1889 zastaralým.

5.1.1 SR: Sender Report RTCP Paket

Sender report paket sestává ze tří sekcí eventuálně následovaných čtvrtou profile-specific rozšířením pokud je definována. První sekce hlavička je 8 oktetů dlouhá. Pole mají následující význam:

version (V): 2 bity

Identifikuje verzi RTP, která je stejná RTCP paketech a RTP datových paketech.

padding (P): 1 bit

Pokud je nastaven padding bit, tento individuální RTCP paket obsahuje určité dodatečné padding oktety na konci, které nejsou částí řídící informace, ale jsou zahrnuty v délce pole. Poslední oktet v paddingu je počet kolik padding oktetů se může ignorovat včetně sebe sama (to bude násobek čtyř). Padding může být potřebný pro určité kódovací algoritmy s pevnou velikostí bloku. Ve složeném RTCP paketu je padding vyžadován pouze v jednom individuálním paketu, protože složený paket je kódován jako celek. Čili padding musí být přidán pouze do posledního individuálního paketu a jestliže je padding do tohoto paketu přidán, musí být padding bit nastaven pouze pro tento paket. Tato konvence pomáhá kontrole validity hlavičky a umožňuje detekci paketů z některých dřívějších implementací, které nesprávně nastavovaly padding bit pro první individuální paket a přidávaly padding k poslednímu individuálnímu paketu.

reception report count (RC): 5 bitů

Počet reception report blocks obsažených v tomto paketu. Nulová hodnota je validní.

packet type (PT): 8 bitů

Obsahuje konstantu 200 k identifikaci jako RTCP SR paketu.

length: 16 bitů

Délka tohoto RTCP paketu v 32-bitovém slovu minus jedna, včetně hlavičky a jakéhokoliv paddingu. (Offset jednoho činí nulu validní délkou a zamezuje eventuální nekonečné smyčce při skenování složeného RTCP paketu zatímco počítání 32-bitových slov zamezuje kontrole validity pro násobky 4).

SSRC: 32 bitů

Identifikátor zdroje synchronizace pro původce tohoto SR paketu.

Druhá sekce, sender informace, je 20 oktetů dlouhá a je přítomná v každém sender report paketu. Sumarizuje přenos dat od tohoto sender. Pole mají následující význam:

NTP timestamp: 64 bitů

Indikuje wallclock, kdy byl tento report poslán, aby se mohl použít v kombinace s timestamps vrácenými ve zprávách příjemce od ostatních příjemců pro měření round-trip propagace těchto příjemců. Příjemci by měli očekávat, že přesnost měření timestamp může být omezena mnohem méně než rozlišení NTP timestamp. Nepřesnost měření timestamp není indikována protože nemusí být známá. V systému, který nemá představu o wallclock time, ale má určitý systém-specific clock jako je „systém uptime“ může sender použít tento clock jako referenci pro výpočet relativního NTP timestamp. Je důležité zvolit obvykle používaný clock, aby v případě, že se použijí separátní implementace k produkci individuálních streamů multimediálních relací, všechny implementace používaly tentýž clock. Až do roku 2036 se budou relativní a absolutní timestamps lišit v nejvyšším bitu takže (neplatné) srovnání ukáže velký rozdíl; poté se očekává, že už nebude potřeba relativních timestamps. Sender, který nemá představu o wallclock nebo uplynulém čase může nastavit NTP timestamp na nulu.

RTP timestamp: 32 bitů

Odpovídá stejnému času jako NTP timestamp (výše), ale ve stejné jednotce a se stejným random offsetem jako RTP timestamps v datových paketech. Tato shoda může být použita pro intra- a inter- média synchronizaci u zdrojů jejichž NTP timestamps jsou synchronizovány a mohou být použity média-nezávislými příjemci k odhadu nominální RTP clock frekvence. Povšimněte si, že v mnoha případech se tento timestamp nebude rovnat RTP timestamp v kterémkoliv sousedním datovém paketu. Spíše se musí vypočítat z odpovídajícího NTP timestamp použitím vztahu mezi RTP timestamp počitadlem a reálným časem, který je udržovaný periodickou kontrolou wallclock time ve vzorkovacím okamžiku.

sender's packet count: 32 bitů

Celkový počet RTP datových paketů přenesených senderem od začátku přenosu až do okamžiku kdy byl vygenerován tento SR paket. Počet by měl být resetován pokud sender změní svůj SSRC identifikátor.

sender's octet count: 32 bitů

Celkový počet payload oktetů (např. bez zahrnutí hlavičky nebo paddingu) přenesených v RTP datových paketech senderem od začátku přenosu až do okamžiku kdy byl vygenerován tento SR paket. Počet by měl být resetován pokud sender změní svůj SSRC identifikátor. Toto pole se může použít k odhadu průměrné payload datové rychlosti.

Třetí sekce obsahuje žádné nebo více reception report blocks podle počtu dalších zdrojů slyšených tímto senderem od posledního reportu. Každý reception report block zprostředkovává statistiku příjmu RTCP paketů z jednotlivého zdroje synchronizace. Příjemci by neměli přenášet statistiky když zdroj změní svůj SSRC identifikátor kvůli kolizi. Tyto statistiky jsou:

2

SSRC_n (source identifier): 32 bitů

SSRC identifikátor zdroje, kterému náleží informace v tomto receptio nreport block.

fraction lost: 8 bitů

Frakce RTP datových paketů ze zdroje SSRC_n lost od doby odeslání předchozího SR nebo RR paketu vyjádřená jako fixní číslo s binární tečkou na levé straně pole. (Toto je ekvivalentní celočíselné části po násobení ztracené frakce 256.) Tato frakce je definovaná jako počet ztracených paketů vydělených počtem očekávaných paketů jak je definováno v dalším odstavci. Pokud je ztráta negativní kvůli duplikátům, fraction lost je nastavená na nulu. Povšimněte si, že příjemce nemůže říct zda kterékoliv pakety byly ztraceny po posledním přijatém a že pro zdroj nebude žádný reception report block vystaven jestliže všechny pakety z tohoto zdroje poslané během posledního reporting intervalu byly ztraceny.

cumulative number of packets lost: 24 bitů

Celkový počet RTP datových paketů ze zdroje SSRC_n, které byly ztraceny od začátku příjmu. Toto číslo je definované jako počet očekávaných paketů mínus počet přijatých paketů, kde počet přijatých paketů zahrnuje všechny zpožděné nebo duplikátní. Počet očekávaných paketů je definovaný jako počet extended last sequence number received jak je definováno dále minus initial sequence number received.

extended highest sequence number received: 32 bitů

Nejnižších 16 bitů obsahuje highest sequence number received RTP data paketu ze zdroje SSRC_n, a nevýznamnějších 16 bitů rozšiřuje toto sekvenční číslo s odpovídajícím počtem sequence number cycles. Povšimněte si, že různí příjemci během stejné relace budou generovat různá rozšíření pro sekvenční čísla pokud se jejich startovací časy významně liší.

interarrival jitter: 32 bitů

Odhad statistické odchylky RTP datového paketu interarrival time měřený v timestamp jednotkách a vyjádřený jako celé číslo bez znaménka. Interarrival jitter J se definuje jako střední odchylka (vyhlazená absolutní hodnota) neshody D v rozestupu paketů u příjemce v porovnání s odesilatelem pro pár paketů. Jak vyplývá z níže uvedené rovnice, je to ekvivalentní neshodě v „relative transit time“ pro dva pakety; relative transit time je neshoda mezi timestamp RTP paketu a hodinami příjemce v okamžiku přijetí měřený ve stejných jednotkách.

Jestliže je Si RTP timestamp paketu i a Ri je okamžik přijetí v jednotkách RTP timestamp pro paket i, pak pro dva pakety i a j může být D vyjádřeno jako

$$D(i,j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Ri - Si)$$

Interarrival jitter by měl být propočítáván průběžně s každým příchodem i data paketu ze zdroje SSRC_n, použitím této neshody D pro tento paket a předcházející paket i-1 v pořadí přijetí (nemusí být v sekvenci) podle tohoto vzorce

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16$$

Kdykoli je vystaven reception report aktuální hodnota J je vzorkována.

Výpočet jitter se musí shodovat se zde specifikovaným vzorcem, aby na profilu nezávislý monitoring mohl provést validní interpretaci reportů přicházejících z různých implementací. Tento algoritmus je optimální pro prvotní odhad v řadě a parametr zisku 1/16 dává dobrý poměr potlačení šumu současně s udržováním rozumné rychlosti konvergence.

last SR timestamp (LSR): 32 bitů

Prostředních 32 bitů ze 64 v NTP timestamp přijatých jako část nejposlednějšího RTCP sender reportu (SR) paketu ze zdroje SSRC_n. Pokud ještě nebyl přijat žádný SR paket, pole je nastaveno na nulu.

delay since last SR (DLSR): 32 bitů

Zpoždění vyjádřené v jednotkách 1/65536 sekund mezi přijetím posledního SR paketu ze zdroje SSRC_n a posláním tohoto reception report block. Pokud ještě nebyl přijat žádný SR paket, je DLSR pole nastaveno na nulu.

Nechť SSRC označuje příjemce vystavením tohoto reportu. Zdroj SSRC_n umí vypočítat round-trip propagation zpoždění k SSRC_r zaznamenáním času A kdy byl tento reception report block přijatý. Vypočítá celkový total round-trip time A-LSR použitím posledního SR timestamp (LSR) pole a pak odečte toto pole k ponechání round-trip propagation zpoždění jako (A - LSR - DLSR). Časy jsou zobrazeny hexadecimálním znázornění 32-bitového pole a ekvivalentním znázornění v plovoucí čárce. Dvojtečky indikují 32-bitové pole rozdělené do 16-bitového integeru a 16-bitové frakce.

Toto může být použito jako přibližné měřítko vzdálenosti příjemců clusterů ačkoliv některé spojení mají velmi asymetrické zpoždění.

5.1.2 RR: Receiver Report RTCP Packet

Formát receiver report (RR) paketu je stejný jako SR paketu, kromě toho, že pole typu paketu obsahuje konstantu 201 a pět slov ze sender information je vynecháno

(jsou to NTP a RTP timestamps a pakety a octet counts sendera). Zbývající pole mají stejný význam jako pro SR paket.

Prázdný RR paket (RC = 0) musí být umístěn ve hlavě složeného RTCP paketu pokud neprobíhá žádný přenos dat nebo report reception.

5.1.3 Rozšíření Sender a Receiver Reports

Profil by měl definovat profil-specifické rozšíření pro sender report a receiver report pokud existuje dodatečná informace, kterou je potřeba pro sender nebo receivers pravidelně reportovat. Tato metoda by se měla použít přednostně pro definování dalších RTCP typů paketů, protože vyžaduje menší režii:

- méně oktetů v paketu (žádná RTCP hlavička nebo SSRC pole);
- jednodušší a rychlejší parsing, protože aplikace běžící pod tímto profilem byly programovány tak, aby vždy očekávaly extension fields v přímo přístupné lokaci po reception reports.

Toto rozšíření (extension) je čtvrtou sekcí v sender – nebo receiver – report paketu, který přichází na konci po reception report blocks pokud tam nějaké jsou. Pokud je vyžadována dodatečná sender information pak by to bylo obsaženo nejdříve v extension sekci, ale pro receiver reports by to nebylo přítomno. Pokud má být zahrnuta i informace o receivers tyto data by měly mít strukturu jako pole bloků paralelní k existujícímu poli reception reports blocks; to znamená, že počet bloků by byl indikovaný RC polem.

5.1.4 Analýza Sender a Receiver Reports

Očekává se, že reception quality feedback bude užitečný ne jen pro sendera, ale také pro ostatní receivers a monitoring třetí strany. Sender může modifikovat své přenosy založené na zpětné vazbě; receivers mohou stanovit zda jsou problémy lokální, regionální nebo globální; network managers mohou použít profil-nezávislé monitory,

kteří přijímají pouze RTCP pakety a ne odpovídající RTP data pakety k ohodnocení výkonu jejich sítí pro multicast distribuci.

Kumulativní county jsou použity jak v sender information a receiver report blocks takže odlišnosti mezi jakýmikoliv dvěma reporty mohou být vypočítány pro provedení měření během krátké a dlouhé časové periody a k poskytnutí odolnosti vůči ztrátě reportu. Rozdíl mezi posledními dvěma přijatými reporty se může použít k odhadu poslední kvality distribuce. NTP timestamp je obsažen takže rychlosti se mohou vypočítat z těchto rozdílů během intervalu mezi dvěma reporty. Protože tento timestamp je nezávislý na clock rate pro kódování dat je možné implementovat na kódování a profilu nezávislé quality monitors.

Příkladem výpočtu je rychlost ztráty paketů během intervalu mezi dvěma reception reports. Rozdíly v kumulativním počtu ztracených paketů dávají počet ztrát během tohoto intervalu. Rozdíl v extended last sequence numbers received dává počet paketů očekávaných během intervalu. Poměr těchto dvou je podíl paketové ztráty během intervalu. Tento poměr by se měl rovnat podílu ztrát pole jestliže jsou tyto dva reporty po sobě jdoucí jinak ne. Rychlost ztrát za sekundu může být získána vydělením podílu ztrát rozdílem v NTP timestamps, vyjádřeným v sekundách. Počet přijatých paketů je počet očekávaných paketů minus počet ztracených. Počet očekávaných paketů může také být použit k posouzení statistické validity jakýchkoliv odhadnutých ztrát. Například, 1 z 5 ztracených paketů má menší váhu než 200 z 1000.

Monitor třetí strany může ze sender information vypočítat průměrnou payload data rate a průměrný packet rate během intervalu bez přijímání dat. Podíl těchto dvou dává průměrnou payload size. Pokud se předpokládá, že paketová ztráta je nezávislá na velikosti paketu pak počet paketů přijatých určitým příjemcem krát průměrná payload size (nebo odpovídající packet size) dává zřejmou propustnost dostupnou danému příjemci.

Kromě kumulativních counts, které umožňují dlouhodobé měření paketových ztrát pomocí rozdílů mezi reporty, fraction lost field poskytuje krátkodobé měření ze single reportu. To nabývá na důležitosti s tím jak se zvětšuje relace do té míry, že příjem

stavových informací nemůže být udržován pro všechny receivers nebo interval mezi reporty se stává tak dlouhým, že pouze jeden report může být přijatý od určitého příjemce.

Interarrival jitter field poskytuje druhé krátkodobé měření přetížení sítě. Paketová ztráta vyznačuje trvalé přetížení zatímco jitter měření vyznačuje přechodné přetížení. Jitter měření může indikovat přetížení ještě dříve než dojde ke ztrátě paketů. Interarrival jitter field je jediným snímkem jitter v čase reportu a není jeho záměrem být brán kvantitativně. Spíše je určený pro srovnání napříč počtem reportů od jednoho příjemce během času nebo od více příjemců, např. v single síti ve stejný čas. Aby bylo umožněno srovnání napříč příjemci je důležité, aby jitter bylo vypočítáno podle stejného vzorce všemi příjemci.

Protože výpočet jitter je založený na RTP timestamps, které zastupují okamžik kdy se vzorkovaly první data v paketu, jakákoliv odchylka ve zpoždění mezi vzorkovacím okamžikem a časem přenosu paketu ovlivní výsledný jitter, který se počítá. Taková odchylka ve zpoždění by se objevila u audio paketů s proměnnou délkou. Také se objeví u kódování videa, protože timestamp je stejný pro všechny pakety v jednom frame, ale tyto pakety nejsou přenášeny najednou. Proměna ve zpoždění do té doby než přenos zredukuje přesnost jitter výpočtu jako měřítko chování vlastní sítě, ale je vhodné to zahrnout, protože uvažujeme, že příjmový buffer to musí akomodovat. Když je výpočet jitter použitý jako porovnávací měřítko (konstanta) komponent kvůli odchylce ve zpoždění do té doby než se přenos odečte takže pak může být pozorována změna v network jitter komponentě do té doby než je relativně malá. Pokud je změna malá pak je pravděpodobné, že bude bezvýznamná.

5.2 CtrMcast

CtrMcast je odvozený od Mcast protokolu. Pro každý uzel musí být vytvořen jeden CtrMcast agent. Je zde také centrální CtrMcastComp agent, který počítá a instaluje multicast cesty pro veškerou topologii. Každý CtrMcast agent zpracovává členské dynamické příkazy a přesměrovává CtrMcastComp agenta k přepočtu vhodných cest. CtrMcastComp je agent počítající centralizovanou multicast cestu.

Metody CtrMcast jsou:

$ctrmcastcomp switch-treetype group-addr – přepne z Rendezvous Point rooted shared tree na source-specific trees pro skupinu specifikovanou group-addr. Tato metoda nemůže být použita k přepnutí z source-specific trees zpátky na shared tree pro multicast skupinu.

$ctrmcastcomp set_c_rp node-list – nastaví RP

$ctrmcastcomp set_c_bsr node0:0 node1:1 – nastaví BSR, určen jako list uzlu: priorita

$ctrmcastcomp get_c_rp node group – navrátí RP pro skupinu, kterou uzel vidí jako multicast skupinu s adresami group-addr. Různé uzly mohou ve skupinách vidět různé RP, jestliže je sít rozdělena stejně jako uzly mohou být v různých úsecích.

$ctrmcastcomp get_c_bsr node – navrátí aktuální BSR pro skupinu

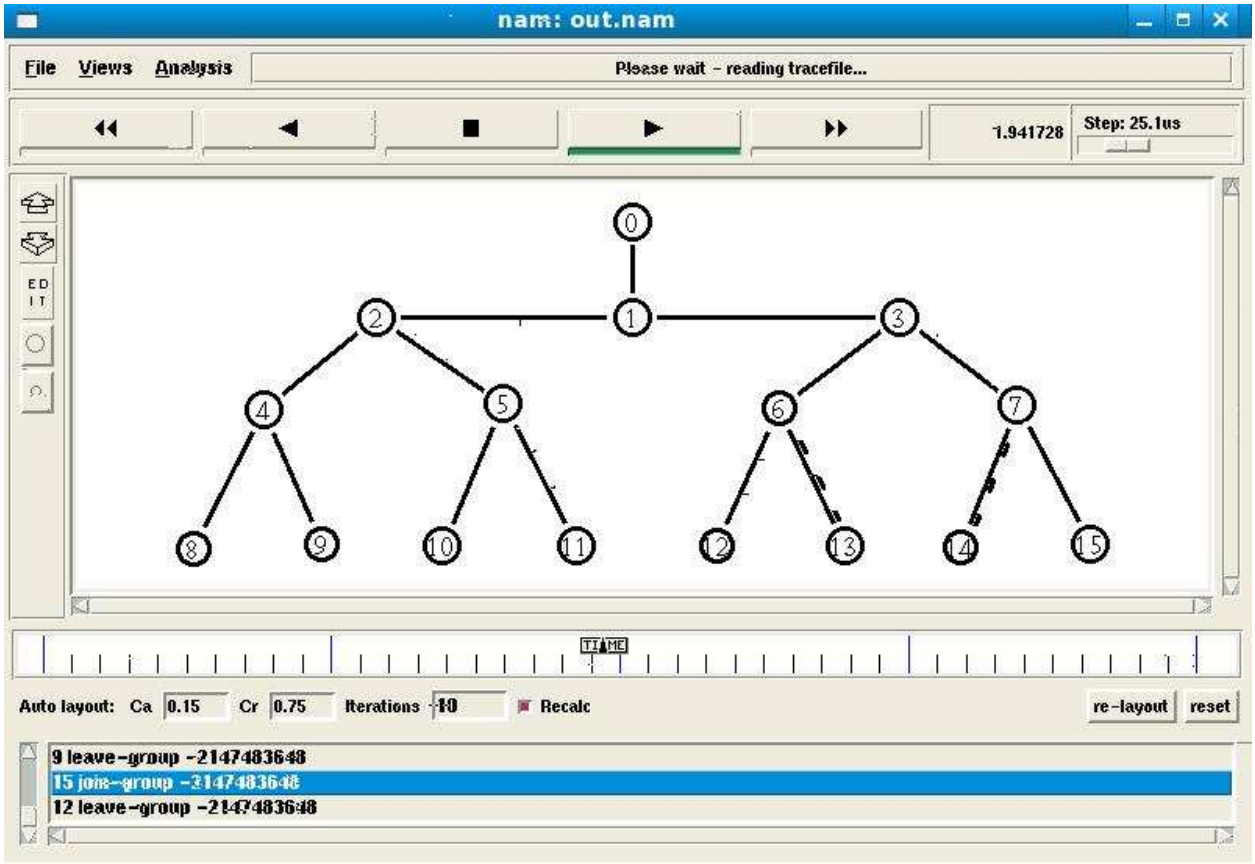
$ctrmcastcomp compute-mroutes – přepočte multicast cesty, jestliže se změní dynamicky sít nebo unicast cesty.

5.3 Popis modelu

Na obr.1 je vidět topologii navrženého modelu sítě. Model se skládá celkem ze šestnácti uzlů. Uzel označený jako 0 představuje zdroj dat. Uzly 8 – 15 představují koncové uživatele, kteří se postupně připojují ke zdroji. Ostatní uzly slouží pouze jako propojovací. Uživatelé se postupně připojují a odpojují od zdroje a ten jim s využitím metody CtrMcast posílá multicast pakety.

Průběh simulace můžeme sledovat nezávisle na simulačním nástroji NS2 v grafickém nástroji NAM, kde je vidět topologie navržené sítě. Jednotlivé toky dat je možné zobrazit po skupinách paketů, které jsou generovaných příslušnými zdroji dat.

V tomto prostředí si můžeme simulaci přehrávat třeba i pozpátku, měnit rychlost simulace a sledovat cesty jednotlivých paketů. Simulace se dá rovněž úplně zastavit a tak si můžeme nechat pohodlně zobrazit detailní informace o každém jednotlivém paketu, který právě putuje sítí.



Obr. 1: *Znázornění simulace v prostředí NAM*

Ukázka zdrojového kódu:

```
# přidělení multicastové adresy;
set group [Node allocaddr]

# vytvoření uzlů
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
```

```
set n9 [$ns node]
set n10 [$ns node]
set n11 [$ns node]
set n12 [$ns node]
set n13 [$ns node]
set n14 [$ns node]
set n15 [$ns node]

# vytvoření linek mezi uzly
$ns duplex-link $n0 $n1 1000Mb 10ms DropTail
$ns duplex-link $n1 $n2 100Mb 10ms DropTail
$ns duplex-link $n1 $n3 100Mb 10ms DropTail
$ns duplex-link $n2 $n4 100Mb 10ms DropTail
$ns duplex-link $n2 $n5 100Mb 10ms DropTail
$ns duplex-link $n3 $n6 100Mb 10ms DropTail
$ns duplex-link $n3 $n7 100Mb 10ms DropTail
$ns duplex-link $n4 $n8 10Mb 10ms DropTail
$ns duplex-link $n4 $n9 10Mb 10ms DropTail
$ns duplex-link $n5 $n10 1Mb 10ms DropTail
$ns duplex-link $n5 $n11 10Mb 10ms DropTail
$ns duplex-link $n6 $n12 10Mb 10ms DropTail
$ns duplex-link $n6 $n13 1Mb 10ms DropTail
$ns duplex-link $n7 $n14 1Mb 10ms DropTail
$ns duplex-link $n7 $n15 1Mb 10ms DropTail

# konfigurace multicast protokolu
set mproto CtrMcast
set mrthandle [$ns mrtproto $mproto]

# vytvoření přijímače
set rcvr [new Agent/LossMonitor]

# definice časů pro připojení a odpojení uživatelů k vysílači
$ns at 0.3 "$n8 join-group $rcvr $group"
$ns at 0.5 "$n9 join-group $rcvr $group"
$ns at 0.7 "$n10 join-group $rcvr $group"
$ns at 0.8 "$n11 join-group $rcvr $group"
$ns at 0.8 "$n12 join-group $rcvr $group"
$ns at 1.2 "$n8 leave-group $rcvr $group"
$ns at 1.3 "$n13 join-group $rcvr $group"
$ns at 1.9 "$n9 leave-group $rcvr $group"
$ns at 1.6 "$n14 join-group $rcvr $group"
$ns at 1.9 "$n10 leave-group $rcvr $group"
$ns at 2.3 "$n15 join-group $rcvr $group"
$ns at 3.5 "$n12 leave-group $rcvr $group"

# definice začátku a konce simulace
$ns at 0.2 "$src1 start"
$ns at 2.0 "$src2 start"
```

```
$ns at 4.0 "finish"

proc finish {} {
	global ns
	$ns flush-trace
	exec nam out.nam &
	exit 0
}
# spuštění simulace
$ns run
```

6 Závěr

Protokol RTP slouží k distribuci a příjmu multimediálních dat v reálném čase a je nezávislý na typu přenosové sítě a protokolu. V projektu jsem se snažil především teoreticky přiblížit co je to multicast a jaké využívá protokoly. Dále je zde seznámení se simulačním prostředím NS2, jeho možnostmi simulace multicastu, porovnání a především bližší představení norem RFC 1889 z roku 1996 a RFC 3550 z roku 2003. Dále je zde seznámení s protokolem RTP/RTCP pro přenos multimediálních dat a především představení paketů RR a SR.

V NS2 jsem navrhl topologii sítě, na které jsem zkoušel simulovat multicast a nakonec jsem použil CtrMcast. Pro přenos RTP a RTCP paketů je v NS2 využívána norma RFC 1889, ale dnes se již běžně v praxi používá norma nová RFC 3550. Tuto novou normu se mi bohužel nepodařilo implementovat do NS2, jelikož se mi nepodařilo správně nastavit přenos RTP/RTCP paketů v simulátoru.

7 Literatura

[1] Information Sciences Institute, *"The Network Simulator - ns-2"*, [online], June 2004, dostupné z: <http://www.isi.edu/nsnam/ns/>

[2] M.A. SPORCTAK, *Směrování v sítích IP*, ComputerPress, a.s., 2004, ISBN: 80-251-0127-4

[3] KOMOSNÝ D., *Nové směry vývoje protokolu RTP/RTCP pro rozsáhlé konference v Internetu*, [online], 2004, dostupné z: <http://www.elektrorevue.cz/clanky/04052/index.html>

[4] LHOTKA L., NOSKA M., *Technologie internetu: Multicast*, [online], 2007, dostupné z: <http://www.computerworld.cz/cw.nsf/id/lhotka_internet_12>

[5] IETF, RFC Pages, RTP*: A Transport Protocol for Real-Time Applications*, [online], January 1996, dostupné z: <http://www.ietf.org/rfc/rfc1889.txt?number=1889>

[6] IETF, RFC Pages, *RTP: A Transport Protocol for Real-Time Applications*, [online], July 2003, dostupné z: <http://www.ietf.org/rfc/rfc3550.txt?number=3550>

Příloha

Zdrojový kód:

```
set ns [new Simulator]
$ns multicast

set f [open out.tr w]
$ns trace-all $f
$ns namtrace-all [open out.nam w]

$ns color 1 red
# prune/graft packets
$ns color 30 purple
$ns color 31 bisque
# RTCP reports
$ns color 32 green

# allocate a multicast address;
set group [Node allocaddr]


set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
```

```
set n9 [$ns node]
set n10 [$ns node]
set n11 [$ns node]
set n12 [$ns node]
set n13 [$ns node]
set n14 [$ns node]
set n15 [$ns node]



#Create links between the nodes
$ns duplex-link $n0 $n1 1000Mb 10ms DropTail
$ns duplex-link $n1 $n2 100Mb 10ms DropTail
$ns duplex-link $n1 $n3 100Mb 10ms DropTail
$ns duplex-link $n2 $n4 100Mb 10ms DropTail
$ns duplex-link $n2 $n5 100Mb 10ms DropTail
$ns duplex-link $n3 $n6 100Mb 10ms DropTail
$ns duplex-link $n3 $n7 100Mb 10ms DropTail
$ns duplex-link $n4 $n8 10Mb 10ms DropTail
$ns duplex-link $n4 $n9 10Mb 10ms DropTail
$ns duplex-link $n5 $n10 1Mb 10ms DropTail
$ns duplex-link $n5 $n11 10Mb 10ms DropTail
$ns duplex-link $n6 $n12 10Mb 10ms DropTail
$ns duplex-link $n6 $n13 1Mb 10ms DropTail
$ns duplex-link $n7 $n14 1Mb 10ms DropTail
$ns duplex-link $n7 $n15 1Mb 10ms DropTail



# configure multicast protocol;
set mproto CtrMcast
# all nodes will contain multicast protocol agents;
set mrthandle [$ns mrtproto $mproto]

# set RV and bootstrap points
```

```
$mrthandle set_c_rp $n0
# $mrthandle set_c_bsr $n(1):0 $n(3):1


set udp1 [new Agent/UDP]
set udp2 [new Agent/UDP]

$ns attach-agent $n1 $udp1
$ns attach-agent $n2 $udp2

set src1 [new Application/Traffic/CBR]
$src1 attach-agent $udp1
$udp1 set dst_addr_ $group
$udp1 set dst_port_ 0
$src1 set random_ false


set src2 [new Application/Traffic/CBR]
$src2 attach-agent $udp2
$udp2 set dst_addr_ $group
$udp2 set dst_port_ 1
$src2 set random_ false

# create receiver agents
set rcvr [new Agent/LossMonitor]

# joining and leaving the group;
$ns at 0.3 "$n8 join-group $rcvr $group"
$ns at 0.5 "$n9 join-group $rcvr $group"
$ns at 0.7 "$n10 join-group $rcvr $group"
$ns at 0.8 "$n11 join-group $rcvr $group"
$ns at 0.8 "$n12 join-group $rcvr $group"
$ns at 1.2 "$n8 leave-group $rcvr $group"
$ns at 1.3 "$n13 join-group $rcvr $group"
```

```
$ns at 1.9 "$n9 leave-group $rcvr $group"
$ns at 1.6 "$n14 join-group $rcvr $group"
$ns at 1.9 "$n10 leave-group $rcvr $group"
$ns at 2.3 "$n15 join-group $rcvr $group"
$ns at 3.5 "$n12 leave-group $rcvr $group"

$ns at 0.2 "$src1 start"
$ns at 2.0 "$src2 start"

$ns at 4.0 "finish"

proc finish {} {
        global ns
        $ns flush-trace
        exec nam out.nam &
        exit 0
}

$ns run
```