



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

SDL MODEL PRO SOURCE SPECIFIC MULTICAST

SDL MODEL FOR SOURCE SPECIFIC MULTICAST

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. STANISLAV ZÁŇ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILAN ŠIMEK

BRNO 2008

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Stanislav Záh
Bytem: Hlavní 125, 79399, Osoblaha
Narozen/a (datum a místo): 14.9.1983, Krnov

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 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: SDL model pro Source Specific Multicast

Vedoucí/školicitel VŠKP: Ing. Milan Šimek

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

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 užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování 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 tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

ANOTACE

Tato práce se zabývá problematikou síťové komunikace s použitím metody Source Specific Multicast. Zaměřuje se na problematiku přihlašování, odhlašování a administrace klientů v multicastové skupině a protokolu IGMP, který má toto na starost. Dále se zaměřuje na problematiku signalizace mezi zdrojem dat a přihlášenými klienty, kteří data odebírají. V úvodu této práce je rozebrána problematika multicastové komunikace. Následuje kapitola se zaměřením na metodu Source Specific Multicast (SSM). Další kapitola se zaměřuje na rozbor používaných protokolů potřebných k distribuci datového vysílání pomocí SSM ve směru od zdroje vysílání ke klientům a opačně. Následující kapitola se zabývá signalizací komunikace v SSM, zvláště se zaměřuje na použití metody přeposílání paketů a metody spojování paketů ve zdroji. Kapitola také uvádí základní matematické vztahy pro výpočet odesílání signalizace a navrhuje také možná řešení a postupy pro zjednodušení komunikace a zmenšení zpoždění, které je pro signalizaci důležitou vlastností. Následně už se práce zabývá rozdíly obou používaných metod signalizace. Na znalostech získaných teoretickými poznatky je v praktické části navržena aplikace simulující komunikaci klientů a zdroje v SSM. Daná problematika je vysvětlena v MSC diagramech. Aplikace také simuluje reálné chování komunikace za použití obou signalizačních metod v závislostech na počtu klientů a použité metodě signalizace. V poslední kapitole jsou výsledky simulace za použití obou signalizačních metod zhodnoceny.

Klíčová slova: klienti, zdroj, signalizace, přeposílání, spojování

ABSTRACT

This work deals with questions of IP net communication, with using method Source-Specific Multicast. It focused on questions of registration, unregistration and administration of clients in multicast group and IGMP protocol, which is made for this communication. Work deals also with problems of signalization between source of data and clients of multicast group. In the introduction of this work questions of communications in multicast are analysed. It is followed by the chapter focused on the specific method of multicast – Source Specific Multicast (SSM). Next chapter is based on the protocols in SSM, which are used for distribution of data stream in source to clients direction and also in the reverse direction. The next chapter deals with signalization of communication in SSM. It specializes for reflection and summarization methods, which are used here. This chapter also shows basic mathematics formulas for sending signalization packets and proposes other solutions and ways for simplify communication and minimize delay, which is for signalization very important. After that, the work deals with differences between these two methods of signalization. The applications builded in practical part of work from the knowledge of theory from previous chapters and it simulates the real communication between data source and clients situated to multicast group. This communication is explained in MSC diagrams. The application also simulates both used methods of signalization and real count of clients in multicast group. The results of simulation are interpreted in the last part of work.

Key words: clients, source, signalization, reflection, summarization

BIBLIOGRAFICKÁ CITACE

ZÁŇ, S. *SDL model pro Source Specific Multicast*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 65 s. Vedoucí diplomové práce Ing. Milan Šimek.

PROHLÁŠENÍ

Prohlašuji, že diplomovou práci na téma „SDL model pro Source Specific Multicast“ 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 následku 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

.....

PODĚKOVÁNÍ

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

V Brně dne

.....

Abecední seznam zkratk

ASM	<i>Any Source Multicast</i> - multicastová technologie
CBT	<i>Core Based Trees</i> - multicastová směrovací architektura
DoS	<i>Denial of Service</i> - útoky na komunikaci v síti
DVMP	<i>Distance Vecteded Monitoring Protocol</i> - multicastový směrovací protokol
FIFO	<i>First In First Out</i> - typ vstupní fronty
IETF	<i>Internet Engineering Task Force</i> - vývojové a podporující internetové standardy
IGMP	<i>Internet Group Management Protocol</i> - protokol pro administraci klientů
IP	<i>Internet Protocol</i> , specifikuje formát paketů a adresová schémata.
IPTV	<i>Internet Protocol Television</i> - televizní vysílání prostřednictvím internetu
ISDN	<i>Integrated Services Digital Network</i> - mezinárodní komunikační standart pro posílání hlasu, videa a dat přes digitální telefonní linky.
ITU	<i>International Telecommunication Union</i> - mezinárodní telekomunikační svaz pro vývoj standardů
MSC	<i>Message Sequence Chart</i> - diagram popisující přechod mezi stavy systému
MSDP	<i>Multicast Source Discovery Protocol</i> - protokol sloužící k identifikaci zdrojů vysílání v muticastové síti
Pid	<i>Process Identifier</i> - předdefinovaný datový typ jazyka SDL.
PIM	<i>Protocol Independent Multicast</i> - směrovací protokol
PIM-DM	<i>Protocol Independent Multicast-Dense Mode</i> - specifikace módu protokolu
PIM-SM	<i>Protocol Independent Multicast-Sparse Mode</i> – mód protokolu PIM
QoS	<i>Quality of Service</i> - síťové služby zaručující předepsanou úroveň komunikace
RP	<i>Rendezvous Piont</i> – specifický směrovač v multicastové síti
RPF	<i>Reverse Path Forwarding</i> - technologie šíření paketů v síti
RR	<i>Receiver Report</i> - typ paketu pro přenos signalizace
RSI	<i>Receiver Summary Information</i> - rozšiřující blok paketu XR
RTCP	<i>Real Time Control Protocol</i> - kontrolní protokol pro výměnu dat v IP síti
RTP	<i>Real Time Protocol</i> - protokol pro výměnu dat v IP síti
SDL	<i>Specification and Description Language</i> - programovací jazyk
SDL CASE	<i>Specification and Description Language Computer Aide Software Engineering</i> - nástroj pro tvorbu modelů v jazyce SDL
SDL/GR	<i>Specification and Description Language/ Graphical Reprasetation</i> - grafická verze jazyka
SDL/PR	<i>Specification and Description Language/ Phrasel Reprasetation</i> - textová verze jazyka
SR	<i>Sender Report</i> - typ paketu pro přenos signalizace
SSM	<i>Source Specific Multicast</i> - modifikace multicastové technologie
TCP	<i>Transmission Control Protocol</i> - hlavní protokol v TCP/IP sítích
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> - soubor komunikačních protokolů používaných při připojení účastníka do internetu.
TTL	<i>Time To Live</i> - hodnota životnosti paketu
UDP	<i>User Datagram Protocol</i> - hlavní protokol ze skupiny internetových protokolů
UML	<i>Unified Modelling Language</i> - standard sekvenčních diagramů
URL	<i>Uniform Resource Locators</i> - specifikace dokumentu v internetu
XR	<i>Extended Report</i> - rozšiřující paket metody spojování paketů

Abecední seznam nejdůležitějších symbolů

PS_{SR}	velikost paketu Sender Report
PS_{RR}	velikost paketu Receiver Report
RR	Receiver Report paket
SR	Sender Report paket
T_{SR}	časový interval pro odesílání paketu SR
T_{RSI}	časový interval pro odesílání paketu XR
T_{RSIMAX}	maximální hodnota časového intervalu pro odesílání paketu XR
T_{RSIMIN}	minimální hodnota časového intervalu pro odesílání paketu XR
T_{RR}	časový interval pro odesílání paketu RR
T_{RRMAX}	maximální hodnota časového intervalu pro odesílání paketu RR
T_{RRMIN}	minimální hodnota časového intervalu pro odesílání paketu
XR	Extended Report paket metody summarization

OBSAH

Seznam obrázků	14
Seznam Tabulek	15
Cíl projektu	16
Úvod	17
1. Multicast obecně.....	18
1.1 Principy šíření paketů v IP síti.....	18
1.1.1 Unicast a broadcast	18
1.1.2 Multicastové šíření paketů.....	18
1.2 Multicastové skupiny	19
1.3 Multicastové stromy.....	19
1.3.1 Zdrojový strom.....	19
1.3.2 Sdílený strom	20
2. Source Specific Multicast – SSM.....	21
2.1 Obecný popis SSM	21
2.2 Nároky kladené na příjemce SSM	21
2.3 Nároky kladené na směrovače.....	22
2.4 Vlastnosti SSM – výhody a nevýhody	22
3. Používané protokoly	23
3.1 Internet Group Management Protocol – IGMP	23
3.1.1 IGMPv1	23
3.1.2 IGMPv2	24
3.1.3 IGMPv3	25
3.2 Protocol Independent Multicast – PIM	26
3.2.1 Dense mode	26
3.2.2 Sparse mode.....	27
3.2.3 Dence-Sparse mode.....	28
3.3 Protokol RTP/RTCP	28
3.3.1 Real Time Protocol	28
3.3.2 Real Time Control Protocol - RTCP.....	29
3.3.2.1 Sender Report - SR.....	30
3.3.2.2 Receiver Report - RR.....	32
4. Multimédální spojení s RTP a RTCP	33
4.1 Komunikační kanály pro RTP a RTCP.....	33
4.2 Metody vytvoření unicastového signalizačního kanálu.....	34
4.2.1 Metoda přeposílání paketů ve zdroji - Reflection.....	34
4.2.2 Metoda spojování paketů ve zdroji - Summarization	34
4.2.3 Hierarchická agregace	35
4.2.4 Signalizační intervaly pro posílání RR A SR paketů	37
4.2.5 Zhodnocení vlastností technologií přeposílání a spojování paketů	37
5. Teoretický návrh simulačního modelu	38
5.1 Komunikace zdroj – klient, protokol IGMPv3.....	38
5.2 Komunikace zdroj – klient, protokol RTP/RTCP.....	38
5.2.1 Signalizace – metoda reflection	38
5.2.2 Signalizace – metoda sumarization.....	39
5.3 Výpočet intervalů pro odesílání RR a SR paketů - Reflection.....	40
5.4 Výpočet intervalů pro odesílání RR a SR paketů - Summarization	42
5.5 Podstatné hodnoty signalizace RTCP	46
6. Návrh Aplikace simulující multicastové spojení	47

6.1	Úvod do SDL.....	47
6.1.1	Historie jazyka SDL.....	47
6.1.2	Oblast aplikací a výhody jazyka SDL.....	47
6.1.3	Formy prezentace.....	48
6.1.4	Základní vlastnosti SDL.....	48
6.1.5	Statický popis jazyka SDL.....	48
6.1.5.1	<i>Systém</i>	48
6.1.5.2	<i>Blok</i>	49
6.1.5.3	<i>Proces</i>	49
6.1.5.4	<i>Procedura</i>	49
6.1.6	Popis dynamického chování jazyka SDL.....	49
6.1.7	Komunikace mezi procesy.....	50
6.1.8	Čas v jazyce SDL.....	50
6.1.9	Adresování v jazyce SDL.....	51
6.2	Realizace aplikace.....	52
6.2.1	Popis simulačního modelu.....	52
6.2.2	Součásti systému aplikace – blokové schéma.....	53
6.2.2.1	<i>Blok clients</i>	54
6.2.2.2	<i>Blok multicast_network</i>	56
6.2.2.3	<i>Blok sources</i>	58
6.2.3	Spuštění simulačního modelu.....	60
6.2.4	MSC (Message sequence charts).....	62
6.2.4.1	<i>Zahájení vysílání dat a přihlášení účastníka do skupiny</i>	62
6.2.4.2	<i>Query - Report a vysílání signalizace receiver reportů</i>	63
6.2.4.3	<i>Signalizace odesílání sender reportů</i>	64
6.2.4.4	<i>Ukončení spojení klientem a odhlášení klienta</i>	65
7.	Praktické využití simulačního modelu.....	68
7.1	Simulace odesílání časových intervalů.....	68
7.1.1	Porovnání vypočtených a simulovaných hodnot.....	68
7.1.2	Porovnání velikosti odeslaných dat RTCP signalizace pro obě metody....	69
7.1.3	Zhodnocení výsledných simulovaných hodnot.....	71
7.2	Simulace hodnot primárních parametrů QoS.....	71
8.	Závěr.....	72
	Použitá literatura:.....	73

Seznam obrázků

Obr. 1: Příklad multicastového šíření paketů [6]	19
Obr. 2: Zdrojový strom [7]	20
Obr. 3: Sdílený strom [7]	20
Obr. 4: Tvorba SSM kanálů pro vysílání z více zdrojů [5]	21
Obr. 5: Struktura zprávy IGMPv1 [12]	23
Obr. 6: Struktura zprávy IGMPv2 [12]	24
Obr. 7: Zpráva Membership Report [12]	25
Obr. 8: Zpráva Membership Query [12]	26
Obr. 9: Zasazení zprávy RTP do zpráv protokolů nižších vrstev [4]	28
Obr. 10: Struktura protokolu RTP [4]	28
Obr. 11: Hlavička protokolu RTCP [4]	29
Obr. 12: Struktura paketu SR [16]	30
Obr. 13: Struktura paketu RR [16]	32
Obr. 14: Protokol RTP/RTCP v modelu multicastové komunikace SSM s unicastovým signalizačním kanálem [13]	33
Obr. 15: Hierarchická agregace [14]	35
Obr. 16: Dělení sumarizačních uzlů do stromové struktury [14]	36
Obr. 17: IGMP zprávy	38
Obr. 18: Zasílání RR a SR paketů - metoda reflection	39
Obr. 19: Zasílání RR a SR paketů - metoda reflection	40
Obr. 20: Závislost časových intervalů T_{RR} na počtu klientů	41
Obr. 21: Závislost časového intervalu T_{RSI} na počtu klientů metody summarization bez kompresního poměru dat	45
Obr. 22: Závislost časového intervalu T_{RSI} na počtu klientů metody summarization v hierarchické agregaci a s kompresí dat	45
Obr. 23: Porovnání počtu přenesených byte (vyžadované šířky pásma) v jednom signalizačním intervalu pro metody reflection a summarization	46
Obr. 24: Strukturované rozdělení systému	48
Obr. 25: Deklarace a nastavení časovače	50
Obr. 26: Blokové schéma simulačního modelu v jazyce SDL	53
Obr. 27: Procesy tvořící blok <i>clients</i>	54
Obr. 28: Ukázka procesu řešícího komunikaci protokolu IGMPv3 v bloku <i>clients</i>	55
Obr. 29: Ukázka procesu pro posílání reportů v bloku <i>clients</i>	56
Obr. 30: Procesy tvořící blok <i>multicast_network</i>	57
Obr. 31: Ukázka struktury bloku <i>network_management</i> a bloku <i>network</i>	58
Obr. 32: Procesy tvořící blok <i>sources</i>	59
Obr. 33: Struktura procesu <i>source_1</i> a ukázka struktury procedury <i>Reflection</i>	60
Obr. 34: SDL simulátor aplikace	61
Obr. 35: Přihlášení do skupiny a vytvoření SG kanálu	62
Obr. 36: Ukázka MSC diagramu popisujícího žádost Query a odpověď Report a odesílání RR paketu	63
Obr. 37: Odesílání SR paketu s hodnotami QoS	64
Obr. 38: Vyhodnocení RR, analýza hodnot QoS a náprava vysílání	65
Obr. 39: Leave Group proces	66
Obr. 40: Vypršení času pro odpověď na dotaz o odběr dat	67
Obr. 41: Ukládání reálných hodnot časových intervalů do výstupního souboru	68
Obr. 42: Porovnání přeneseného objemu dat za jednotku času	70

Seznam Tabulek

Tab. 1: Hodnoty časových intervalů RR a SR = 0,5Mbps	41
Tab. 2: Hodnoty časových intervalů RR a SR pro B = 1Mbps.....	41
Tab. 3: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR.....	42
Tab. 4: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 1Mbps, bez komprese dat	42
Tab. 5: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 0,5Mbps, s kompresí.....	44
Tab. 6: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 1Mbps, s kompresí	44
Tab. 7: Porovnání reálných a vypočtených hodnot T_{RR} – reflection.....	69
Tab. 8: Porovnání reálných a vypočtených hodnot T_{RSI} – summarization.....	69
Tab. 9: Počet odeslaných paketů během simulace a celková velikost přenesených dat	70

Cíl projektu

Cílem tohoto projektu bylo seznámit se s technologií šíření multimediálních dat multicastovým způsobem, přesněji s její modifikací Source –Specific Multicast. Dále byl stanoven úkol prozkoumat protokolové vybavení komunikačních prostředků ke správnému přenosu dat, kterými jsou Internet Group Management Protocol (IGMP) a směrovací protokoly Protocol Independent Multicast (PIM). Jako hlavní cíl práce byl stanoven rozbor problematiky zasílání signalizace mezi komunikujícími prvky prostřednictvím protokolu RTP/RTCP.

Pro možnost ověření těchto teoretických informací bylo dále úkolem navržení vlastní aplikace realizující komunikaci mezi klienty a zdrojem dat v SSM multicasu a díky ní ověřit matematické modely signalizačních metod přeposílání paketů (reflection) a spojování paketů (summarization).

Úvod

Tématem této diplomové práce je seznámit se s technologií a teorií týkající se multicastových operací, převážně technologie Source Specific Multicast, jejich následný rozbor, nástin problematiky, pomocí jazyka SDL navrhnout aplikaci simulující již zmíněný multicastový přenos a díky ní simulovat reálné výsledky a chování sítě s teoretickým předpokladem. Takto řešenou komunikaci potom následně vysvětlit pomocí ukázek MSC diagramů.

V dnešní době internetu je zcela běžná komunikace založená na bázi vysílání určitého toku dat od jistého zdroje směrem k danému příjemci, respektive skupině příjemců. Takovýmto přenosem dat je tedy myšleno například dnes populární rádiové či televizní vysílání prostřednictvím internetu. V případě takového spojení za účelem přenosu, či vysílání těchto dat od jedné strany ke druhé, či naopak, je nutno znát alespoň nejdůležitější vlastnosti a charakter daného přenosu dat přes IP síť. Díky těmto vlastnostem a znalostem víme, jakou technologii šíření paketů využít a jaký charakter a vlastnosti komunikace zvolit pro úspěšný přenos dat, kvalitní vysílání a spokojenost zákazníků.

Pokud se tedy rozhodneme vysílat ať už rádiové či televizní vysílání přes internet, musíme znát a předpokládat charakter přenosu dat a problémy s ním spojené. Vysílání ovšem musí být nějakým způsobem řízeno a zdroj vysílání musí být informován o kvalitě svého vysílání pro koncové účastníky vysílání. Proto se v tomto spojení zdroj dat a klienti zavádí signalizace, neboli určité kontrolní řízení, která se řeší zpětnou komunikací klienti – zdroj.

V této práci tedy bude naznačena teorie problematiky multicastového přenosu na základě vysílání od zdroje k příjemcům. Bude zde naznačena jak technologie multicast, tak ve velké míře její další specifikace, kterou je Source Specific Multicast, dále už jen SSM, a její další podrobnosti. Spolu s ní i potřebné komunikační protokoly zajišťující správný chod takto zvoleného přenosu dat přes internet a IP síť, jejich struktura, organizace a charakteristika. Budou zde naznačeny protokoly potřebné ke správné komunikaci zdroje a spotřebiče vysílaných dat, jejich vzájemnou výměnu zpráv a veškerá komunikace řazená časově podle reálné komunikace. Dále zde budou uvedeny možnosti signalizace a zpětné komunikace, jejich výhody a nevýhody a celková problematika komunikace zdroj – klienti a naopak klienti – zdroj. Na základě těchto poznatků bude následně v programovém prostředí jazyka SDL navržen simulační model tak, aby objasnil problematiku komunikace na MSC diagramech a dále na základě určitých vstupních dat simuloval danou komunikaci tohoto modelu a potvrdil, vyvrátil, či upřesnil teoretické poznatky získané během zpracovávání této práce.

1. Multicast obecně

Cíl kapitoly: V této kapitole se seznámíme se základními vlastnostmi multicastového šíření dat a vlastnostmi této technologie.

1.1 Principy šíření paketů v IP síti

1.1.1 Unicast a broadcast

Základní principy šíření paketů dat v IP sítích jsou unicast, broadcast nebo multicast. V případě unicastového přenosu se jedná o nejjednodušší způsob komunikace, ve kterém je jeden zdroj dat a jeden jeho příjemce. Adresace v síti při přenosu paketů nemusí řešit žádné složitosti, protože pakety posílané v síti mají v sobě uvedenou jak zdrojovou, tak cílovou adresu a samostatně síť procházejí. Směrovač na základě těchto adres doručí pakety k jeho cíli.

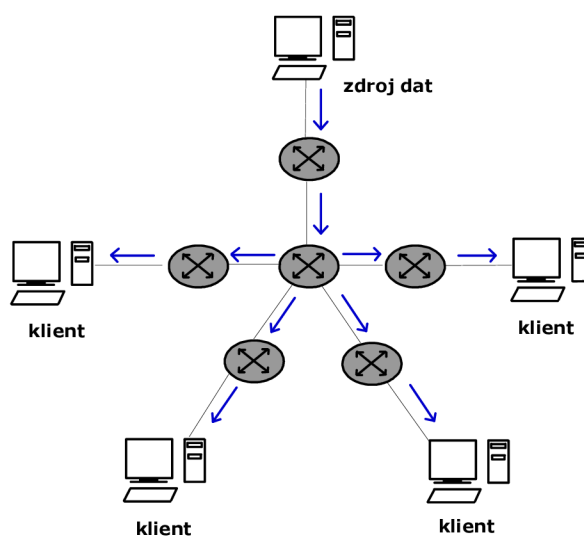
Z hlediska broadcastového přenosu se jedná o vysílání dat od jednoho zdroje do všech systému na síti. Tento přenos využívá některých protokolů pro mapování cesty sítí, ale toto vysílání také dokáže být docela zneužitelné a to tak, že některé servery mohou síť zahlcovat daty, která nikdo nepotřebuje, ani si je nikdo nevyžádal. Pro broadcast je v IP sítích vyhrazena adresa 255.255.255.255.

1.1.2 Multicastové šíření paketů

Pro multicastové šíření paketů je typické, že zdroj dat (může to být i skupina), posílá stejná data velkému počtu příjemců. Pro názornost je tedy ideální již zmíněné internetové vysílání televize či rádia. Jestliže máme zdroj dat vysílající tato data, směrovač nebo skupinu směrovačů, která tyto data po síti správně rozesílá a skupinu příjemců, kteří tyto data obdrží, pak komunikace po síti může vypadat následovně. Klient, který chce vysílaná data obdržet, se musí nejprve přihlásit do multicastové skupiny. Směrovač ho ve skupině zaregistruje, a po dalším nastavení už danému klientovi posílá data ze zdroje. O této skupině se více dozvíme později, kde bude podrobněji probrána. Zdroj dat tyto data vysílá pouze jednou, o duplikaci a distribuci těchto dat se starají směrovače, které ovšem musí tento multicast podporovat. Takto je ošetřeno zbytečné zatěžování sítě přebytečnými daty, protože data se duplikují jen tam, kde je to potřebné. To klade velkou náročnost na funkci směrovačů, které také musí ošetřit, zda data neduplikují zbytečně a uživatel přijímající tato data je nedostává zbytečně ve více kopiích.

Nevýhodou ale stále zůstává to, že se jedná o nespolehlivý přenos dat, který ve velkém případě využívá UDP protokol. V případě ztráty těchto paketů tedy nelze očekávat jejich opětovné preposlání. Pokud se ale jedná o data jako jsou video či hlas, není to až takový problém. Ztrátovost se projeví jenom na kvalitě audio či video signálu.

Příklad multicastového šíření paketů zobrazuje obr. 1.



Obr. 1: Příklad multicastového šíření paketů [6]

1.2 Multicastové skupiny

Adresace používaná v multicastových skupinách je typu D, a to tedy odpovídá rozsahu 224.0.0.0 – 239.255.255.255. Jednotlivá rozhraní ale nemají přidělenou adresu, což je rozdílné od ostatních adres, zde totiž adresy označují jakousi početně nedefinovanou skupinu uživatelů.

Známé skupiny jsou tedy například 224.0.0.0/23, což je skupina pro směrování a údržbu. Dalším je rozsah adres od 239.192.0.0 do 239.251.255.255, což je privátní prostor adres pro jednu organizaci, a adresy od 239.252.0.0 do 239.255.255.255, které určují privátní prostor adres pro jednu lokalitu.

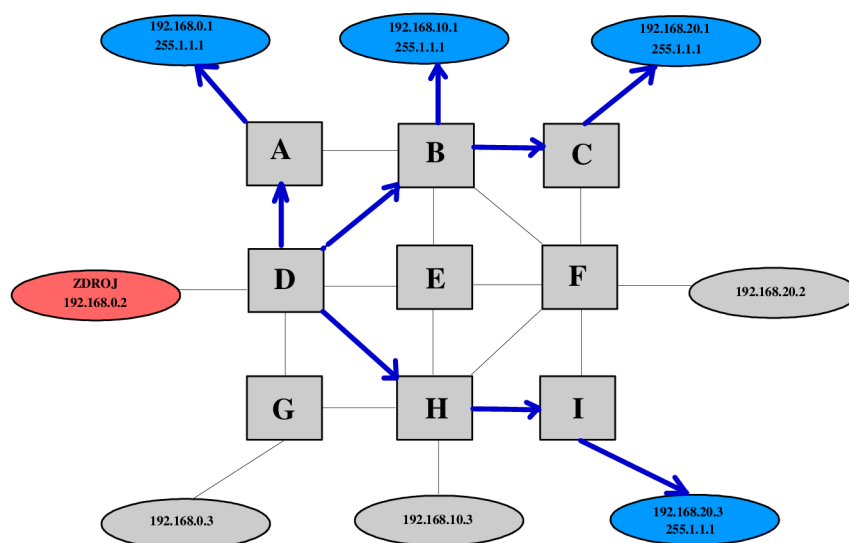
Multicastová adresa je určena pouze pro cíl, proto se nesmí za žádné situace objevit ve zdrojové adrese paketu. Multicastový vysílací kanál je obecně specifikován notací (*, G), kde G je IP adresa skupiny a hvězdička značí zdroj dat, jehož IP adresu neznáme.

1.3 Multicastové stromy

Pro pochopení jak multicast funguje je třeba zmínit také teorii multicastových stromů.

1.3.1 Zdrojový strom

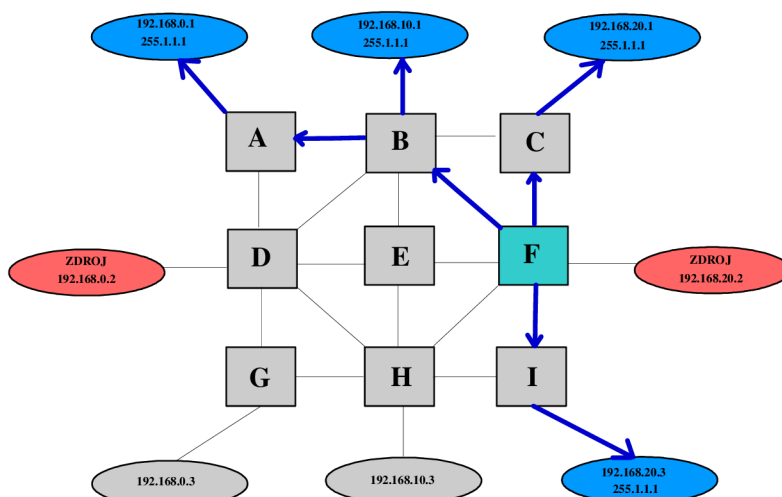
Zdrojový strom, jinak také zvaný strom nejkratších cest, má v kořenu zdroj multicastových dat a jeho listy obsahují příjemce vysílání. Strom se označuje notací (S,G), kde S značí adresu zdroje a písmeno G označuje adresu multicastové skupiny. Příklad takového stromu zobrazuje obrázek č.2. Z obrázku je tedy zřejmé, že pro různé zdroje posílající data do stejné multicastové skupiny existují různé zdrojové stromy. Někdy se také tento způsob práce s multicastem označuje jako Source Specific Multicast (SSM).



Obr. 2: Zdrojový strom [7]

1.3.2 Sdílený strom

Sdílený strom má n rozdíl od zdrojových stromů kořen vždy na jednom místě, nezávisle na tom, kdo data posílá. Tomuto kořenu se obvykle říká rendezvous point (RP), a proto se také tyto stromy někdy nazývají RP stromy. Jako označení pro tyto stromy se používá notace (*, G). Hvězdička označuje, že strom není závislý na zdroji multicastu. Jak je vidět, směrovač F tvoří kořen stromu. Situace však není jednoduchá a tyto stromy jsou dvojího druhu: jednosměrné a obousměrné. Distribuce multicastu v obousměrném stromu probíhá tak, že zdroj data vysílá směrem ke kořenu, a zároveň po směru stromu k listům. U jednosměrného stromu se data pošlou unicastem ke kořenu, a ten se o jejich distribuci již postará. Tento způsob chápání multicastu má také označení ASM (Any Source Multicast).



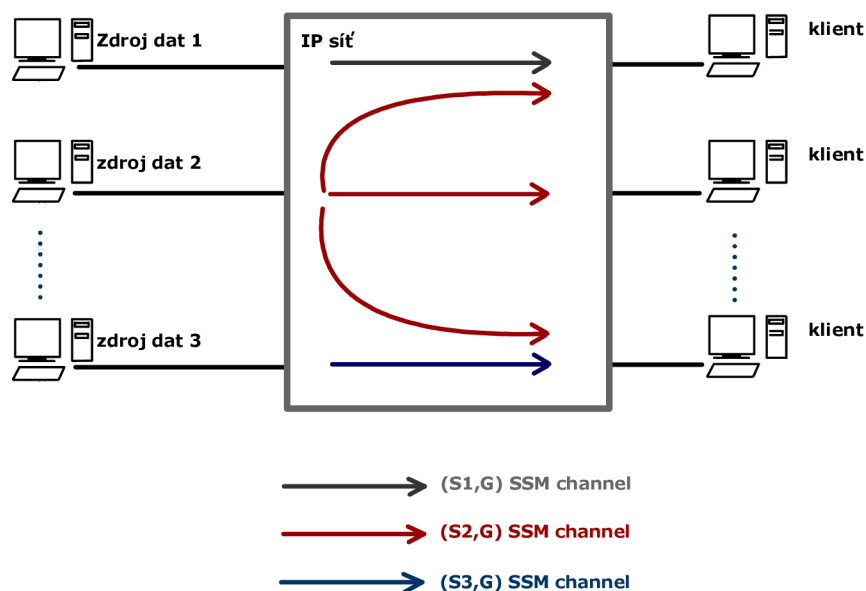
Obr. 3: Sdílený strom [7]

2. Source Specific Multicast – SSM

Cíl kapitoly: Seznámit se s obecnými vlastnostmi tohoto speciálního druhu multicastu.

2.1 Obecný popis SSM

SSM je pokročilejší technologie vycházející s klasického multicastu, využívajícího kanál (*, G). Má pro sebe vyhrazený speciální adresový prostor v rozsahu od 232.0.0.0 - 232.255.255.255. Pokud se tedy uživatel rozhodne přijímat data prostřednictvím této technologie, musí zadat nejbližšímu směrovači nejen multicastovou adresu skupiny G, ale i unicastovou adresu zdroje S. Každé vysílání je tedy určeno adresou zdroje a adresou multicastové skupiny (S, G). Tímto je možné, aby různé zdroje vysílaly do jedné multicastové skupiny, a přitom nedošlo ke konfliktu. V SSM technologii se takové multicastové skupině říká SSM kanál (SSM channel). Tuto problematiku osvětluje obrázek níže. Tím je ošetřen problém s alokací adres a nedostatečným řízením přístupu. Ovšem do sítí, kde není žádný příjemce, data být vysílána nesmí.



Obr. 4: Tvorba SSM kanálů pro vysílání z více zdrojů [5]

2.2 Nároky kladené na příjemce SSM

Mezi nároky kladené na příjemce SSM patří nutnost zajištění **rozšíření funkčního rozhraní IP modulu příjemce**. To znamená, že pokud protokoly zažádají o příjem datagramů, které jsou odeslány do příslušného (S, G) kanálu, musí jim to umožnit síťové rozhraní. S tím také souvisí i žádost o ukončení vysílání těchto paketů v (S, G) kanálu. Tuto problematiku zajišťují 4 parametry: **socket** (umožňuje rozlišovat různé služby žádající o vysílání), **zdrojová IP adresa**, **multicastová IP adresa skupiny** a **síťové rozhraní příjemce**.

Dále také musí být zajištěno **rozšíření modulu příjemce**. Zde se musí zabezpečit, aby datagram, jenž přijde na dané síťové rozhraní, byl doručen jenom těm soketům, kteří si o něj zažádali v registraci (S, G) kanálu.. Těm, co si o něj nezažádali, doručen být nesmí. K registraci a následné deregistraci slouží protokol IGMPv3 (viz. kapitola č. 3). Pokud tento protokol není síťovými prvky podporován, a ty používají pouze IGMPv1/v2, můžeme využít formu od společnosti CISCO IGMP v3lite.

2.3 Nároky kladené na směrovače

Pokud vysíláme síť data multicastem, musí tudíž i směrovače logicky podporovat toto vysílání. To znamená, že musí podporovat protokol PIM (protocol independent multicast) a jeho modifikaci pro SSM. Dále také směrovač řeší přihlašování a odhlašování členů do skupin, tudíž musí také opět podporovat IGMPv3 nebo IGMPv3lite.

Jakmile se sestaví zdrojový strom, je třeba, aby se zajistilo odstranění paketů putujících dokola. K tomu slouží RPF (Reverse Path Forwarding). Dnešní směrovače dovolují používat naráz SSM, a ve zbylém adresovém prostoru také ASM.

2.4 Vlastnosti SSM – výhody a nevýhody

Výhody:

Do jednotlivých multicastových skupin může klidně vysílat více zdrojů, protože každé vysílání si bude udržovat svůj vlastní zdrojový strom, komunikace se mezi sebou nebudou plést a nedojde ke konfliktu či zahlcení datového kanálu daty z cizího zdroje. SSM podporuje také využití jedné multicastové adresy pro více zdrojů, pokud jsou ovšem data posílána různými aplikacím.

Mezi hlavní výhody použití technologie SSM patří její snadná realizace a administrace.

Nemůže dojít k útokům DoS (Denial of Service), protože každý uživatel přijímá data z multicastové skupiny jen od konkrétního zdroje.

SSM lze použít i pro konference s velkým počtem účastníků.

Nevýhody:

Hlavní nevýhodou použití této technologie, ale vlastně i celého multicasu je, že pro jeho přenosy nemůžeme použít bezpečný a potvrzovaný přenos dat protokolem TCP.

Další již výše zmíněnou nevýhodou je, že pokud chceme v síti provozovat SSM, musí na to být všechny prvky v síti připraveny, a musí tedy podporovat protokol IGMPv3 nebo alespoň IGMPv3lite.

3. Používané protokoly

Cíl kapitoly: Kapitola se zaměřuje na rozbor nejdůležitějších a nejpoužívanějších protokolů technologie SSM a jejich následný rozbor.

3.1 Internet Group Management Protocol – IGMP

Použití vícesměrového vysílání IP v sítích TCP/IP je definováno jako standard TCP/IP ve specifikaci RFC 1112, Internet Group Management Protocol (IGMP). Tento protokol navrhl Steve Deering. Využívají jej hlavně klienti signalizující směrovačům své členství v multicastových skupinách. Nicméně to není jen jediné použití. IGMP zprávy jsou posílány v IP datagramech s dobou „životnosti“ Time- To_Live (TTL).

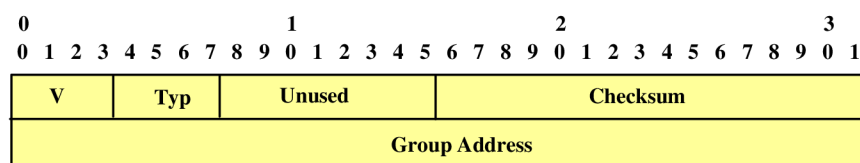
3.1.1 IGMPv1

IGMPv1 je velmi jednoduchý protokol obsahující dvě zprávy:

- **MEMBERSHIP REPORT** – zprávu posílá uživatel jako odpověď na dotaz směrovače, zda-li je na segmentu zájemce o danou skupinu. Zpráva se posílá na adresu multicastové skupiny.
- **MEMBERSHIP QUERY** – zprávu posílá směrovač na každém rozhraní jako dotaz, zda-li je na segmentu nějaký uživatel, který má zájem o některou multicastovou skupinu.

Jakmile klient přestane mít zájem o multicastovou skupinu, přestane na dotaz směrovače odpovídat zprávou Membership Report.

Struktura zprávy je zobrazena na obrázku 5.



Obr. 5: Struktura zprávy IGMPv1 [12]

Kde jednotlivé pole zprávy IGMPv1 znamenají:

(V) - Verze – 4bitové pole nastavené vždy na hodnotu 1

Typ – 4bitové pole, jež značí typ zprávy. Obsahuje dvě hodnoty 1 a 2.

- Typ1 – Host Membership Query (jenom směrovače)
- Typ2 – Host Membership Report (jenom klienti)

(Unused) - Nevyužité – 8bitové pole je nastaveno na nulu při posílání, při odesílání se ignoruje.

(Checksum) - Kontrolní součet – 16bitové pole. Při výpočtu se toto pole vynuluje.

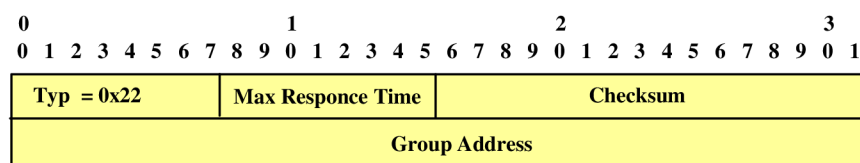
(Group Address) - Skupinová adresa – je implicitně nastavená na 0.0.0.0, když směrovač odesílá Membership Query. Jestliže klient odešle Membership Report, skupinová adresa se nastaví na multicastovou adresu, na kterou se chce připojit.

3.1.2 IGMPv2

Druhá verze IGMPv2 byla vyvinuta z důvodů lepších odhlašovacích mechanismů ze skupiny. Přináší následující změny:

- **LEAVE GROUP messages** – zpráva, kterou se může uživatel explicitně odhlásit z multicastové skupiny. Výhodou je zkrácení doby při odhlášení.
- **GROUP-SPECIFIC QUERY messages** – umožňuje směrovačům posílat dotaz přímo na konkrétní skupiny. Tato zpráva je užitečná, když směrovač přijme zprávu LEAVE GROUP pro nějakou multicastovou skupinu.
- **MAXIMUM RESPONSE TIME field** – nové pole v Query zprávách, které povoluje nastavit směrovači čas odezvy (MRT) pro Host Membership Report.

Strukturu zpráv IGMPv2 zobrazuje následující obrázek.



Obr. 6: Struktura zprávy IGMPv2 [12]

Popis jednotlivých polí ve zprávě IGMPv2.

Typ – narozdíl od předcházející verze protokolu má IGMPv2 k dispozici tři typy zpráv:

- **MEMBERSHIP QUERY** - dva podtypy:
 - **GENERAL QUERY** – zjišťuje, které skupiny mají v síti své členy.
 - **GROUP-SPECIFIC QUERY** – zjišťuje, jestli má konkrétní skupina v síti připojené nějaké členy.
- **Verze 1 MEMBERSHIP REPORT** – používají IGMPv2 klienti pro zpětnou konfiguraci s klienty IGMPv1.
- **Verze 2 MEMBERSHIP REPORT** – posílá člen skupiny, který informuje směrovač, že se minimálně jeden člen skupiny nachází na podsíti.
- **LEAVE GROUP** – posílá člen skupiny, který informuje směrovač o odhlášení ze skupiny.

(Maximum Response Time) - Doba pro odpověď – 8bitové pole, které má význam jenom ve zprávách Membership Query a určuje maximální povolenou dobu pro posílání odpovědí v desetinách sekund. V ostatních případech je hodnota nastavena na nulu.

(Checksum) - Kontrolní součet – 16bitové pole. Při výpočtu se toto pole vynuluje.

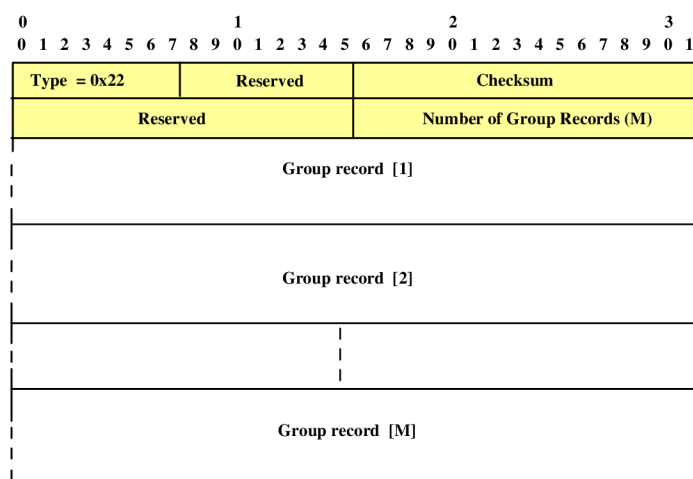
(Group Address) - Skupinová adresa – V případě General Query je nastaveno na nulu, v případě Group-Specific Query je v něm uložena konkrétní adresa. V Membership Report nebo Leave Group je v něm uložena adresa skupiny, do které chce být uživatel připojen nebo kterou chce opustit.

3.1.3 IGMPv3

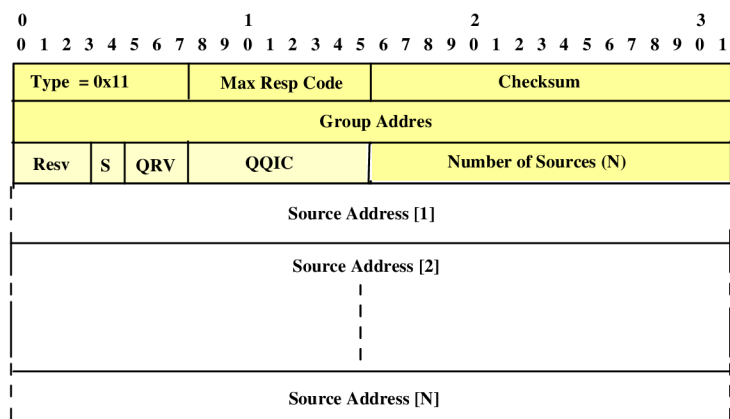
Problém, který se tato verze snaží řešit, spočívá v tom, že pokud jste členem nějaké multicastové skupiny, nechcete často přijímat zprávy od všech, ale pouze od vybraných zdrojů. Proto se v IGMPv3 nepřihlašujete pouze do skupiny (*, G), ale přihlašujete se k odběru dat z konkrétního zdroje v dané skupině (S, G). Přirozeně se výrazně změnil i formát zpráv. Zprávy již nemají konstantní délku, jako tomu bylo u předchozích verzí, ale mají své dvě vlastní zprávy, a pro zpětnou kompatibilitu rozumí dalším třem:

- IGMPv3 Membership Query - kód 11,
- IGMPv3 Membership Report - kód 22,
- IGMPv1 Membership Report - kód 12,
- IGMPv2 Membership Report - kód 16,
- IGMPv2 Leave Group - kód 17.

Na následujících obrázcích jsou zobrazeny formáty zpráv Membership Report a Membership Query.



Obr. 7: Zpráva Membership Report [12]



Obr. 8: Zpráva Membership Query [12]

Vlastní postup vypadá následovně:

Zájemce o příjem dat zašle přesně definovanou zprávu IGMP *Join* nejbližšímu směrovači. Ten ho ve skupině zaregistruje, vytvoří kanál pro přenos dat a posílá zpátky zájemci zprávu *Membership Query* jako dotaz na to, zda-li chce uživatel přijímat data. Ten – má-li zájem, odesílá obratem zprávu *Membership Report* zpět na nejbližší router do té doby, dokud chce data přijímat. Jakmile se uživatel rozhodne ukončit relaci, odešle na router zprávu *Leave Group* o tom, že ukončuje spojení. Router přijme zprávu a odesílá zprávu *Group and Source Specific Query* na skupinu jako dotaz, zda-li je ještě nějaký uživatel ve skupině a má zájem o odběr dat.

Prvních 8 bytů Query zprávy je stejných jako u předchozích verzí, takže klienti s nižší verzí protokolu nepoznají rozdíl a odpoví zprávou *Report* v odpovídající verzi. Router pak zachází s každou skupinou dle verze přihlášených klientů. Pokud se tedy do stejné skupiny na stejné síti přihlásí IGMPv2 i IGMPv3 klient, router nebude provádět filtrování zdrojových adres a bude posílat všechna data skupiny. Naopak pokud klient dostane zprávu délky 8 bytů, ví, že jeho router používá IGMP nižší verze, a přizpůsobí se mu.

3.2 Protocol Independent Multicast – PIM

Pozn.: Níže popsané protokoly řeší pouze směrování, ale nemají na starosti přihlašování do skupin. O to se stará protokol IGMPv3, který je probrán v kapitolách výše. Protokol PIM pracuje v několika režimech. Tyto režimy a následně použití PIM rozebírají následující dvě kapitoly.

3.2.1 Dense mode

PIM Dense Mode protokoly využívá Source Specific Multicast (SSM). Fungují na *push* principu, to znamená, že se provoz „tlačí“ do všech větví stromu, když tam příjemce není. Aby se zabránilo zbytečnému zabírání přenosového pásma, listy stromu, které nemají zájem o vysílání, pošlou směrem ke kořenu tzv. *prune* (anglicky ořezat) zprávu. Větev, ze které zpráva přišla, je oříznuta. Tímto způsobem se pokračuje, dokud nezůstanou jen listy, které data chtějí přijímat. Zpráva *prune* má však jen omezenou platnost, takže pokud listy

stále nechtějí data přijímat, musí vysílání zprávy opakovat, jinak nadřazený směrovač obnoví zaslání dat. Pokud nastane opačný jev, to znamená, že se data do větve nevysílají, ale přihlásí se nový zájemce, směrovač pošle zprávu *graft* (anglicky připojit) a data začnou do nové větve opět téci. Hlavní zástupci jsou PIM-DM, DVMP.

Protocol independent multicast (PIM) - Dense mode

PIM, je, jak název napovídá, protokolově nezávislý. To je myšleno tak, že tento protokol využívá pro své potřeby unicastové směrovací tabulky směrovače a nezáleží mu na tom, jaký protokol tyto tabulky vytvořil. V dense modu je tento protokol vhodný pro využití v rychlých lokálních sítích, kde je většina uživatelů členem nějaké multicastové skupiny. Jak již bylo řečeno, PIM –DM počítá s tím, že je v síti volné široké přenosové pásmo, protože využívá techniku flood and prune (je možno přeložit jako zaplavit a ořezat). Příchozí pakety testuje mechanismem RPF na základě směrovacích tabulek směrovačů.

3.2.2 Sparse mode

Sparse mode protokoly využívají pro distribuci dat sdílené stromy a tzv. *pull* model. Ten spočívá v tom, že se data neposílají do žádné části sítě, která si o ně sama nezažádala. Pokud se připojí zájemce o data, pošle nadřazený směrovač zprávu *Join* ke kořenu. Tím se sestaví nová větev a data mohou proudit k zájemci. Zpráva *Join* má časově omezenou platnost, tudíž musí být obnovována, jinak je přenos dat ukončen. Pokud již ve větvi není žádný člen, směrovač sám pošle do kořene zprávu *graft* a větev se odřízne. Cíl modelu sparse je jasný, má ulehčit zátěž směrovačům. Výhodou oproti modu dense mode je, že směrovače nemusí počítat pro každý zdroj nový strom. Opět ale záleží na tom, jak je umístěn RP, neboť při nevhodném umístění může být cesta paketů krkolomná a zdoluhavá, tudíž neefektivní. Sparse modu se týká ještě jeden problém. Pokud by se měla tato metoda nasadit masově, každý internetový poskytovatel by chtěl mít svůj vlastní RP, aby ho mohl spravovat a mít pod kontrolou. To by ovšem způsobilo, že by RP bylo velké množství a vysílající by se s příjemcem nepotkali. Bylo by tedy třeba nějakým dalším protokolem řídit RP. Příklad takového protokolu je MSDP (Multicast Source Discovery Protocol). Zástupci jsou sparse protokolů jsou PIM-SM, CBT.

Protocol independent multicast (PIM) – Sparse mode

PIM - SM je řízen poptávkou po datagramech, umí využívat oba druhy stromů (zdrojový a sdílený, viz 1.2.1 a 1.2.2). Provoz vychází z RP směrem k příjemcům. Když začne v síti nějaký zdroj S generovat data pro skupinu G, je tento provoz zachycen nejbližším směrovačem a informace o provozu je poslána na RP. RP zkontroluje, zda je pro onu skupinu již vytvořen sdílený strom (*, G). Pokud ne, přihlásí se RP do skupiny G a požádá o vytvoření zdrojového stromu (S, G). Pak již přijímané multicastové datagramy šíří dolů sdíleným stromem (*, G).

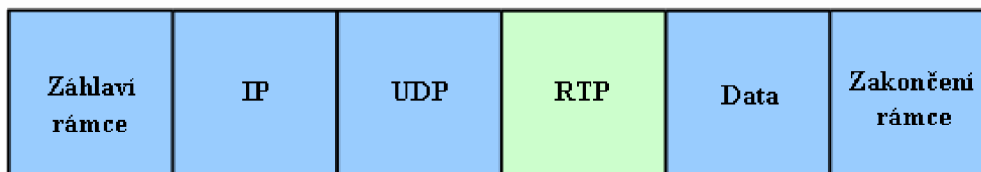
3.2.3 Dence-Sparse mode

Dence-Sparse mód je pouze kombinací výše zmíněných a využívá se v sítích, kde se zároveň využívají jak klasické multicastové přenosy, tak novější SSM. Protokol se sám přepne do modu, který je třeba pro přenos. Tedy pro SSM dense mode, pro multicast sparce mode. Představitel je PIM sparce-dense mode.

3.3 Protokol RTP/RTCP

3.3.1 Real Time Protocol

RTP je zkratka slov Real Time Protocol. Protokol RTP je protokol pro přenos multimediálních dat (hlasu a videa) přes IP síť. Je používán jak v řešeních navržených ITU, tak i IETF. Je navržen tak, aby byl jak stupňovatelný pružný, tak i aby byl oddělen přenos uživatelských dat od řídicích funkcí. Původní standard TCP/IP měl pouze dva transportní protokoly: TCP a UDP. S příchodem multimédií se ukázalo, že tyto dva nestačí, a proto byl vytvořen protokol pro přenos v reálném čase RTP, a k němu řídicí RTCP protokol. RTP je vlastně jakási nadstavba nad UDP (viz obr.6). Oproti UDP zajišťuje navíc doručování paketů ve správném pořadí, (toho se dosahuje pomocí číslování paketů nebo časových razítek).

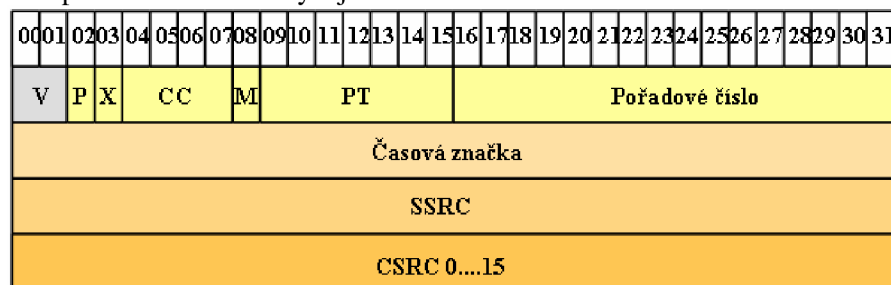


Obr. 9: Zasazení zprávy RTP do zpráv protokolů nižších vrstev [4]

Protokol RTP nese informace o:

- typu kódování informace
- okamžicích hovoru
- identifikaci odesílatele
- detekci ztráty synchronizace
- segmentaci a opětovném skládání
- zabezpečení dat (utajení)

Hlavičku protokolu RTP zachycuje obr. 10.



Obr. 10: Struktura protokolu RTP [4]

Zde jednotlivé informace znamenají:

- V (Version) – verze protokolu - 2 bity
- P (Padding) – informace o přidání výplně, délka výplně je uvedena v posledním bajtu datové části – 1bit
- X (eXtension) – rozšiřovací bit -1 bit
- CC (CSRC) - počet CSRC identifikátorů, které následují za pevnou hlavičkou – 4 bity.
- M (Marker) – bit pro hlasovou komunikaci a videokomunikaci. U hlasové komunikace se bit M využívá pro označení prvního paketu po vynechání.
- PT -Typ dat (Payload type) – kódovací metoda pro audio/video. Velikost 7 bitů.
- Pořadové číslo (Sequence number) – pořadové číslo datového segmentu. 16 bitů.
- Časová značka (Time stamp) – vyjadřuje okamžik vzniku prvního vzorku uvedeného v datové části. Časová značka je nejčastěji vyjádřena násobkem vzorkovací periody. Velikost 32 bitů.
- SSRC (Synchronization Source Identifier) - identifikátor zdroje – náhodné číslo jednoznačně identifikující zdroj.
- CSCR – identifikuje zdroje přispívající do užitečného obsahu. Počet přispívajících zdrojů je určen polem počtu CSRC (CC). Celkem může být 16 přispívajících zdrojů. Jestliže je více přispívajících zdrojů, je výsledný užitečný obsah sloučením těchto zdrojů.

3.3.2 Real Time Control Protocol - RTCP

Protokol RTCP (Real-time Transport Control Protocol) slouží k řízení přenosu informace a signalizaci svázané s přenosem multimédií pomocí protokolu RTP. Protokol opět pracuje nad protokolem UDP. Signalizace je v tomto protokolu přenášena ve dvou typech paketů, a to informace o zdroji dat ve směru od zdroje k příjemci (Source Report - SR), a o příjemci (Receive Report - RR) ve směru od příjemce ke zdroji.

zprávy odesílatele SR: informace o odeslaných datech, synchronizační časové značky.

zprávy příjemce RR: informace o přijatých datech, ztrátách, chybách, zpoždění a jeho rozptylu.

informace o odesílateli: jméno, e-mail, telefon, identifikace.

Hlavičku řídicího protokolu RTCP zobrazuje následující obrázek.

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V	P	Počet				Typ						Délka																			
Data:....																															

Obr. 11: Hlavička protokolu RTCP [4]

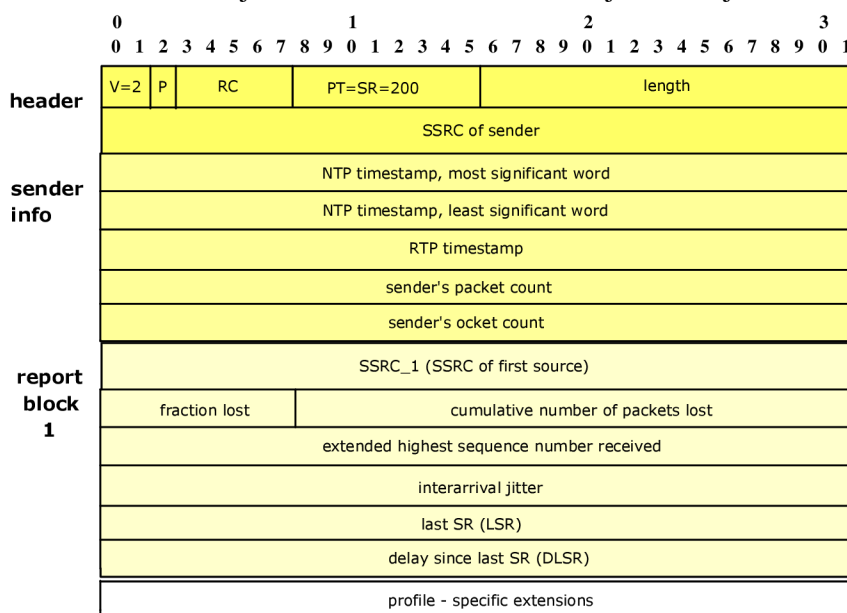
Zde jednotlivé informace znamenají:

- V (Verze protokolu) – musí být stejná jak v RTP paketech tak i v paketech RTCP. Velikost-2 bity.

- P (Padding) - informace o přidání výplně, délka výplně je uvedena v posledním bajtu datové části – 1bit.
- Počet (Count) - číslo bloků přijatých zpráv obsažené v tomto paketu. Velikost 5 bitů.
- Typ – typ RTCP protokolu. Velikost 8 bitů.
- Délka - délka tohoto RTCP balíčku včetně hlavičky a výplně. Vyrovnaní jednoho způsobuje nulovou platnou délkou a vyhne se možné nekonečné smyčce v kontrole složení RTCP paketů.

3.3.2.1 Sender Report - SR

Struktura paketu Sender Report, dále jen SR, je rozdělena do třech hlavních částí. Tyto jsou: záhlaví (Header), informace vysílajícího (Sender Info), a třetí část je report blok (Block Report). Těchto report bloků může mít paket několik, ale jejich struktura je vždy stejná, pouze informace v ní jsou odlišné. Strukturu zobrazuje následující obrázek.



Obr. 12: Struktura paketu SR [16]

Header (záhlaví) je obvykle dlouhé 8 oktetů a jednotlivé položky v něm znamenají:

Verze (V): 2 bity

Identifikuje verzi RTP, která musí být stejná jak v paketech RTCP tak i v RTP.

P (Padding): 1 bit

Jestliže je bit výplně nataven, tak tento individuální RTCP paket obsahuje na konci některé přídavné oktety výplně, které nejsou částí řídicí informace jsou obsaženy v délce pole.

PT (Packet Type): 8 bitů

k identifikaci, že jde o RTCP SR paket, je zvolena konstanta 200.

Length (délka): 16 bitů,

Délkou tohoto SR paketu jsou 32-bitové slova minus jedna, včetně záhlaví a výplně.

SSRC (Synchro Source Identifier): 32 bitů,

identifikátor zdroje je náhodné číslo jednoznačně identifikující zdroj.

Sender info (informace odesílatele), je část paketu přítomná v každém SR paketu a jeho délka je 20 oktetů. Sčítá data odeslaná tímto odesílatelem. Sender info je složena z následujících částí:

NTP timestamp (časová razítka): 64 bitů

Hodnota používaná spolu s časovými značkami RR (Receive Report) ostatních příjemců k výpočtu celkovému zpoždění, neboli „Round Trip Delay“.

RTP timestamp (časová značka): 32 bitů

Zavedena z podobných důvodů NTP timestamp pro značení RTP paketů časovými razítky, pro snadné vyhodnocování přijatých paketů. Toto slouží například k synchronizaci posílání multimediálních dat.

Sender's packet count (počet paketů odeslaných zdrojem): 32 bitů

Celkové číslo paketů odeslaných zdrojem od začátku vysílání až do doby, kdy byl generován tento SR paket. Počet může být vynulován, jestliže odesílatel (zdroj) změní svůj SSRC identifikátor.

Sender's octet count (Počet oktetů odeslaných zdrojem): 32 bitů

Pole může být použito k odhadu průměrné rychlosti přenosu dat RTP.

Třetí část paketu SR obsahuje žádný nebo více „reception“ report bloků. To záleží na množství příjemců, kteří „slyšeli“ zdroj od posledního reportu. Každý report blok přenáší statistiky pro přijetí RTP paketů z jednotně synchronizovaného zdroje. Prvky report bloku jsou:

SSRC_n (identifikátor zdroje): 32 bitů

Identifikátor zdroje, kterému informace v tomto bloku náleží.

Fraction lost (částečná ztrátovost): 8 bitů

Zlomek RTP dat, vyslaných zdrojem o identifikátoru SSRC_n, ztracených od odeslání předchozího SR nebo RR paketu. Tato část je definována jako číslo paketů ztracených ku počtu paketů očekávaných. Když je ztrátovost nulová, je toto pole nastaveno na nulu.

Cumulative number of packets lost (rostoucí číslo ztracených paketů):

24 bitů

Celkové číslo RTP data paketů, které byly ztraceny od začátku vysílání těchto dat. Toto číslo je dáno jako poměr paketů očekávaných ku počtu paketů doopravdy přijatých.

Extended highest sequence number received (číslo nejdelší přijaté sekvence): 32 bitů

Prvních 16 bitů obsahuje hodnotu nejdelší přijaté sekvence RTP dat, a horních 16 bitů slouží pro rozšíření této sekvence.

Interarrival jitter (mezipříchozí kolísání zpoždění): 32 bitů

Kolísání velikosti zpoždění paketů při průchodu sítí (vzniká například na směrovačích jako důsledek změn směrování nebo chování interních front směšovačů.

Last SR timestamp – LSR (poslední časová značka SR): 32 bitů

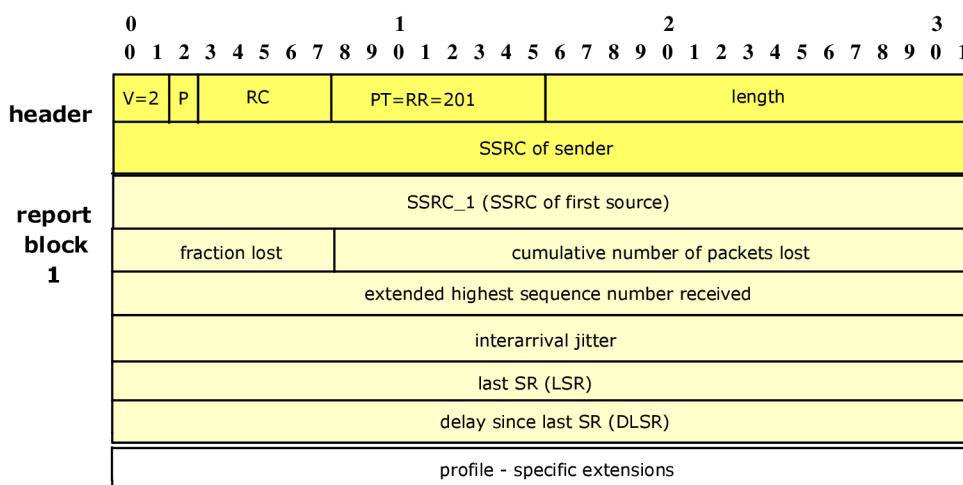
Když ještě nebyl přijatý žádný paket SR, je hodnota nastavena na nulu.

Delay since last SR – DLSR (zpoždění během posledního SR): 32bitů

Zpoždění, vyjádřené v jednotkách 1/65536 sekund, mezi přijetím posledního SR paketu ze zdroje SSRC_n a posláním tohoto report bloku. Když ještě nebyl přijat žádný SR paket, je toto pole nastaveno na hodnotu nula.

3.3.2.2 Receiver Report - RR

Struktura paketu Receiver Report, dále jen RR, je obdobná jako struktura paketu SR s výjimkou, že nenese informace v bloku sender info, a ten je celý proto vypuštěn. Strukturu paketu SR zobrazuje následující obrázek. Hodnoty a vlastnosti polí jsou shodné jako výše popsání stejné hodnoty v paketu SR.



Obr. 13: Struktura paketu RR [16]

Hodnoty v doplňkových blocích „profile – specific extensions“ slouží k definování a rozšíření informací o zdroji dat, jestliže ten potřebuje přidat informace, které potřebuje posílat v souladu s pravidly mezi zdroji a příjemci RTCP paketů. Toto rozšíření může být také například použito s výhodou k definici jiného typu RTCP paketu.

4. Multimedální spojení s RTP a RTCP

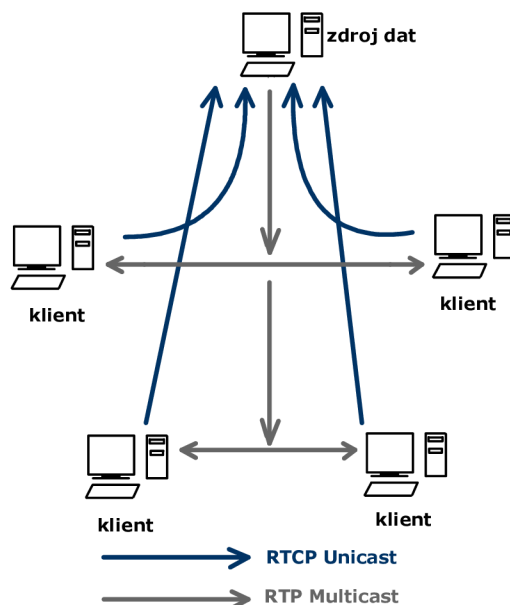
Cíl kapitoly: Nástin problematiky komunikace mezi zdrojem a klienty, její případné řešení.

4.1 Komunikační kanály pro RTP a RTCP

V dnešní době jsou multimedální konference konané prostřednictvím internetu stále oblíbenější. V pořádání těchto konferencí dochází k rozporům, jak přenášet data (RTP) „jeden k mnoho“, a signalizaci (RTCP) „mnoho k mnoho“. Zde totiž vzniká rozdíl mezi použitím ASM nebo SSM. V současné době komerčního internetu dokáže použití SSM danou situaci výrazně zjednodušit.

Problém však nastává tehdy, jestliže SSM poskytuje jenom jednosměrné spojení od zdroje k příjemcům. To znemožňuje jednotlivým klientům multicastové skupiny používat protokol RTCP, jenž vyžaduje komunikaci „příjemce – zdroj“ nebo i mezi samotnými „příjemci“.

Tuto situaci je možné vyřešit použitím dvou možných komunikací, a to pro protokol RTP nezměněný multicast ve směru od zdroje k „příjemcům“. Ovšem pro protokol RTCP zavedeme zpětný unicastový kanál směrem od klienta, „příjemce“, ke zdroji dat. Pro názornost je tato problematika naznačena na obrázku 12. Data, která jsou od „příjemců“ ke zdroji posílána unicastem, jsou zdrojem zpět opět šířena v podobě multicastu.



Obr. 14: Protokol RTP/RTCP v modelu multicastové komunikace SSM s unicastovým signalizačním kanálem [13]

4.2 Metody vytvoření unicastového signalizačního kanálu

Pro zajištění unicastového signálu můžeme zvolit dvě metody. První je metoda přeposílání paketů ve zdroji (Reflection), druhá je metoda spojování paketů ve zdroji (Summarization).

4.2.1 Metoda přeposílání paketů ve zdroji - Reflection

Princip metody přeposílání paketů spočívá v tom, že zdroj přijímá všechny unicastové pakety RR protokolu RTCP od jednotlivých příjemců, které pak nazpět přeposílá díky multicastu všem příjemcům. Jinými slovy se každý přijatý paket RR znovu individuálně odešle zpět všem příjemcům. V tomto případě výhoda tkví v provedení nepatrných změn na straně „příjímačů“. To znamená, že namísto multicastového vysílání RR paketů je použito unicastové. Výhodou je i snadné rozpoznání zdrojů paketů. Z toho pak můžeme jasně určit, který z „příjemců“ má problémy s příjmem dat (chybovost, zpoždění, atd.).

Tato metoda vypadá jednoduše, ale přeposílání jednotlivých paketů od zdroje k příjemcům je neefektivní. Například přenos informací o zpoždění, které jsou důležité jenom pro zdroj vysílání a nejsou tedy nutné pro zpětné vysílání.

4.2.2 Metoda spojování paketů ve zdroji - Summarization

Narozdíl od předchozí metody spočívá tato metoda v tom, že zdroj spojuje unicastově přijatá data v paketech RR do jakéhosi „spojovaného paketu“ (Summary Packet), a ten je následně pomocí multicastu poslán nazpět směrem zdroj -> příjemce všem příjemcům. Slučování ve zdroji se děje díky různým matematickým algoritmům, které umožňují kompresi dat v porovnání s přeposíláním všech paketů jako u předchozí metody. Díky této kompresi tedy metoda „šetří“ přenosové pásmo pro zasílání signalizace. Tato přednost se nejlépe využije při rozsáhlém konferenčním spojení.

Pro tuto metodu byl navrhnut typ paketu protokolu RTCP a to paket Extended Report (XR). XR paket obsahuje blok Receiver Summary Information (RSI) a další volitelné bloky. Blok RSI udává obecné informace a volitelné bloky pak poskytují přesnější specifické hodnoty. Paket XR je zasílán společně se signalizačními pakety SR, a to v přesně stanoveném časovém intervalu pro stanovení signalizace – signalizačním intervalu.

Volitelné bloky XR paketu jsou např:

- **Blok ztrátovosti** - umožňuje příjemci porovnání kvality příjmu (ztrátovosti paketů) s ostatními příjemci.
- **Celková ztrátovost** - narozdíl od bloku ztrátovosti je jeho účelem poskytnout dlouhodobý pohled na kvalitu příjmu.
- **Blok změny zpoždění** - přenáší změnu zpoždění paketů protokolu RTP. Tyto údaje jsou používány pro výpočet dynamických parametrů vyrovnávacích pamětí.
- **Blok zpoždění** - přenáší zpoždění paketů protokolu RTP. Zdroj je jediný člen skupiny, který může zpoždění vyhodnocovat.
- **Unicastová adresa příjímače** - poskytuje unicastovou adresu příjemce pro možnost jejich rozlišení.
- **Blok šířky pásma protokolu RTCP** - informuje příjímače o vyčleněné šířce pásma pro protokol RTCP (pro zdroj i pro příjemce).

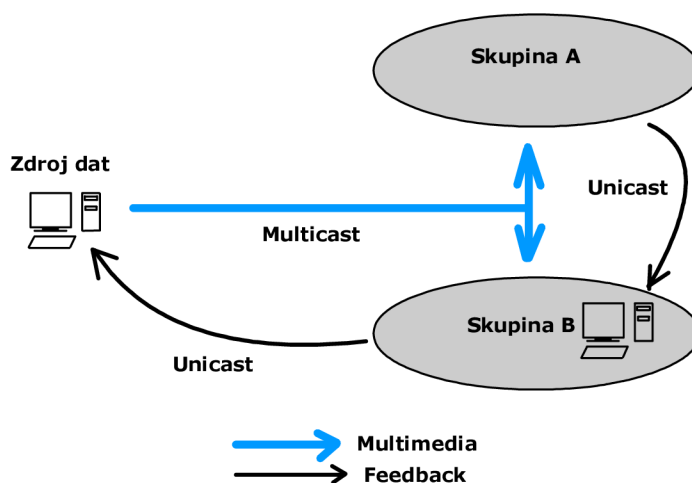
Požadavky na protokol RTCP, který má za úkol vysílat pakety RS a RR jsou, že tyto pakety musí vysílat periodicky v určitém časovém intervalu, již zmíněném signalizačním intervalu, a závisí na počtu účastníků („příjemců“). To závisí také na tom, aby nedocházelo k zabírání šířky pásma určeném pro výměnu vlastních multimediálních dat protokolem RTP.

Pro protokol RTCP je vyhrazeno 5% z celkové přenosové kapacity, a to pro pakety RR je z toho určeno 3, 75%, pro pakety SR zbývajících 1, 25%.

Pozn.: Přenos přeposílaných, popř. spojovaných paketů ze zdroje k příjemcům není započítáván do výpočtu přenosové kapacity. To znamená, že zdroj nezahrnuje tyto pakety jako své vlastní signalizační pakety.

4.2.3 Hierarchická agregace

Hierarchická agregace je rozšířením metody spojování paketů. V této metodě vystupuje člen relace (nebo vyhrazený server) jako spojovací uzel (summarization node) pro skupinu více členů této relace. Tato metoda je zobrazena na následujícím obrázku. Ten popisuje, že klienti obsažení ve skupina A (Group A) posílají zpětnou vazbu skupině B (Group B) a jejich uzel dává obdrženou zpětnou vazbu do jednoduché sumarizace. Toto spojení je posíláno jako zpětná vazba klienta skupiny B. Sumarizační uzly jsou organizovány hierarchicky, tedy od uzlů vytvářejících jednoduchou sumarizaci pro další vyšší sumarizační uzel, a tak dále až po nejvyšší uzel, kterým je zdroj dat.



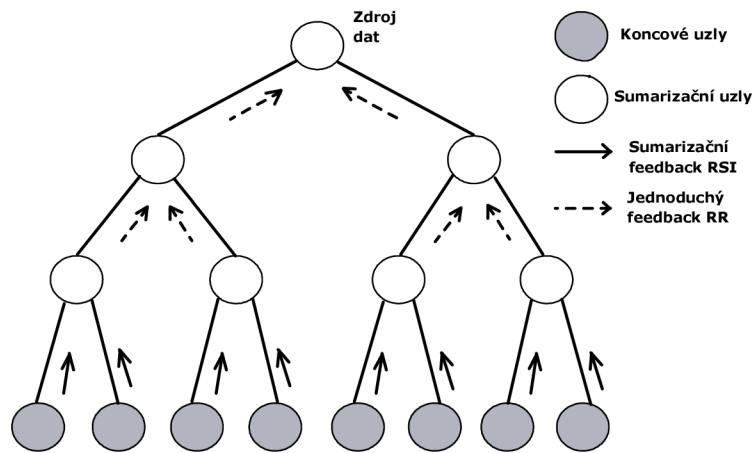
Obr. 15: Hierarchická agregace [14]

Pro pochopení výše popsané situace a metody je na dalším obrázku znázorněna stromová struktura a dělení uzlů. Popsáno bylo, že sumarizační uzly jsou organizovány hierarchicky. To znamená, že sumarizační uzel nejvýše ve stromu je zdroj multimediálních dat. V postupu níže a níže přes sumarizační uzly se dostaneme až na poslední vrstvu sumarizačních uzlů. Tu představují klienti daného relačního spojení, kteří dostávají od zdroje multimediální data, jež jsou vysílána cestou „jeden k mnoha“ multicastovým způsobem. Z obrázku stromové struktury je tedy zřejmé, že zpětná vazba (feedback) je vysílána od koncových uzlů, které využívají RR pakety, do sumarizačního uzlu výše ve větví. Sumarizační uzly spojují přijaté hodnoty ve feedbacku do RSI paketů,

zmíněných v kapitole 4.2.2. Tento RSI paket je poslán dále k nejbližšímu sumarizačnímu uzlu výše ve struktuře. Velikost takto spojovaného paketu se vypočítá podle (1)

$$PS_{RR-RSI} = RR_RSI_{fix} + \sum_{k=1}^K (RBL_{fix} + RBL_{data}(k)), \quad (1)$$

Tento sumarizační uzel spojuje už jenom hodnoty paketů RSI. Tato procedura se opakuje každým krokem směrem ve struktuře výše až do té doby, než sumarizační bod nejvýše ve struktuře, tedy zdroj dat, dostane spojený feedback od všech účastníků relace. Nejvyšší uzel posílá spojený feedback zpátky metodou „jeden k mnoho“ multicastovým způsobem. Následně posílá zdroj multimediálních dat jeho vlastní feedback všem členům relace v paketech SR.



Obr. 16: Dělení sumarizačních uzlů do stromové struktury [14]

Pro tuto zobrazenou strukturu lze potom vypočítat signalizační interval zpětné vazby relace jako:

$$RI_{SS} = RI_{end \rightarrow smr(I-1)} + \sum_{i=I-1}^1 RI_{smr(i) \rightarrow smr(i-1)} + RI_{smr(0) \rightarrow mem}, \quad (2)$$

kde $i \geq 0$ je úroveň stromu, kde hodnota 0 náleží pozici zdroje a hodnota $i = I$ je úroveň koncových uzlů. Značení $RI_{end \rightarrow smr(I-1)}$ je signalizační interval od koncových uzlů k sumarizačnímu uzlu, $RI_{smr(I) \rightarrow smr(i-1)}$ je signalizační interval od sumarizačního uzlu k nejbližšímu výše postavenému sumarizačnímu uzlu a $RI_{smr(0) \rightarrow mem}$ je signalizační interval od zdroje dat zpět k samotným členům relace.

Jak již bylo popsáno výše, rozložení šířky pásma pro posílání signalizace RTCP je celkově 5%. Z těchto 5% je 3,75% vyhrazeno pro paketu RR a zbývajících 1,25% je vyhrazeno pro SR pakety. Tudíž můžeme uvést vztah, podle kterého můžeme vypočítat signalizační interval RR paketu od koncových uzlů k nejbližšímu sumarizačnímu uzlu jako :

$$RI_{end \rightarrow smr(I-1)} = RI_{RR} = \frac{PS_{RR} \times n_{gend}}{0,75 \times BW_{RTCP}}, \quad (3)$$

Kde PS_{RR} je velikost RR paketu a n_{gend} je počet koncových uzlů náležícímu sumarizačnímu uzlu.

4.2.4 Signalizační intervaly pro posílání RR A SR paketů

Jakmile máme v architektuře SSM jeden jediný zdroj dat a n_T klientů, kteří tato data dostávají, službu alokující šířku pásma BW_T a délku paketu PL_{RR} , můžeme vypočítat periodu odesílání RR paketů podle (4) jako :

$$T_{RR} = \frac{PL_{RR} \times n_T}{0,0375 \times BW_T}, \quad (4)$$

Vysílací interval paketu SR vysílaného od zdroje ke všem klientům je dán podle (5) takto:

$$T_{SR} = \frac{PL_{SR} \times n_T}{0,0125 \times BW_T}, \quad (5)$$

Kde PL_{SR} je délka paketu SR.

4.2.5 Zhodnocení vlastností technologií přeposílání a spojování paketů

Metoda přeposílání paketů: s rostoucím počtem „příjemců“ (účastníků) roste také počet paketů, a tedy i celkový počet bajtů zasílaných v jednom signalizačním intervalu. Je tudíž zabírána větší přenosová kapacita. Velikost přeposlaného paketu se nijak nemění od původní velikosti paketu RR. Pro SSM je velikost stanovena na 92 byte, viz (15). Analogicky lze tedy vyjádřit, že výsledný počet byte d , čili velikost paketů posílaných během jednoho signalizačního intervalu při počtu účastníku n , uvádí následující vztah (6).

$$d = 92 \times n, \quad (6)$$

Metoda je vhodná pro méně rozsáhlé spojení o malém počtu účastníků, kde není potřeba ovlivňovat frekvenci vysílání signalizační údajů. Kdybychom měli rozsáhlé spojení s velkým počtem účastníků, narůstá s ním i signalizační interval, kdy mohou daní účastníci vysílat. Některé informace od příjemců mohou být již příliš staré a nelze je použít pro další zpracování. Z toho důvodu jsou v případě velkých konferencí informace přijaté během jednoho intervalu považovány spíše za odhad než přesné hodnoty. Metoda spojování paketů se snaží tento nepříznivý vliv eliminovat.

Metoda spojování paketů: Díky spojování paketů, kde jsou informace z RR paketů slučovány a potom posílány v rámci jednoho paketu, nám vychází nelinearita počtu byte d v závislosti na počtu účastníků n , kde musíme brát v úvahu ještě kompresi dat m ($m = 1 - 16$). Ta nám udává kolik stupňů sumarizace, tedy počty sumarizačních bodů, které dané spojení má, viz. obr. 14. Výsledný počet byte, čili velikost paketu pro sumarizaci, posílaný během jednoho signalizačního intervalu se dá vypočítat podle (7).

$$d = 76 + 96 \log_2((n/m) + 1), \quad (7)$$

Z výpočtu je jasné, že dochází k úspoře vyžadované šířky přenosového pásma díky redukcí počtu byte přenesených v jednom signalizačním intervalu, kde je posílán pouze jeden paket. Celkovou šířku pásma tedy udává velikost jednoho paketu.

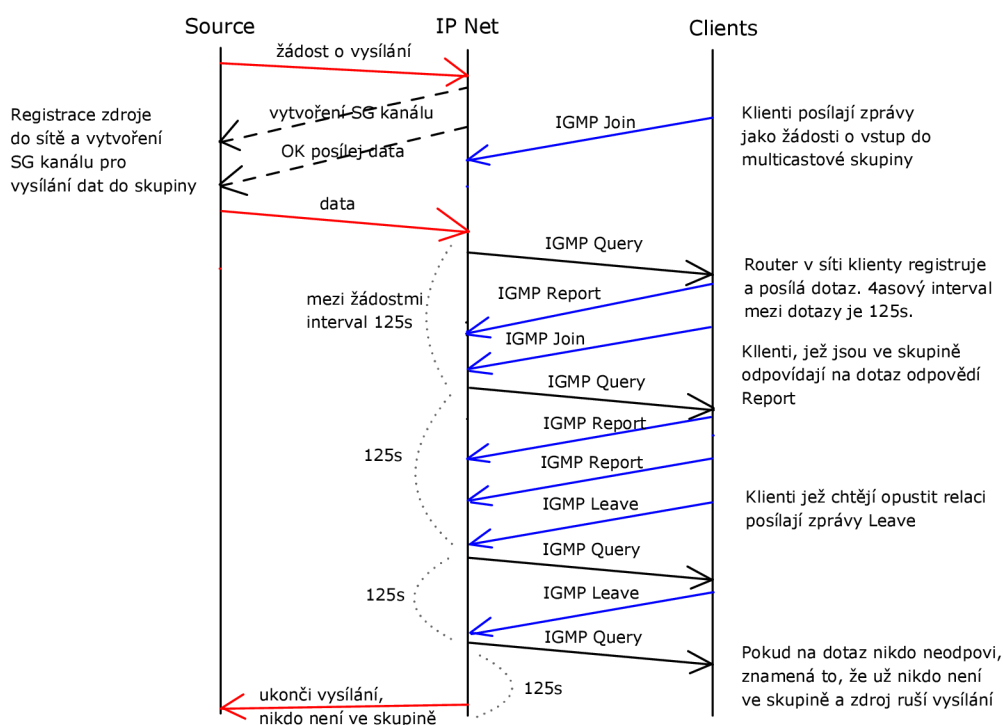
Výhody této metody spočívají v úspoře šířky pásma pro přenos signalizace, zvýšení počtu přenášených signalizačních informací (např. mohou být přidány informace popisující vlastnosti příjemců), a zvýšení frekvence zasílání signalizačních informací.

5. Teoretický návrh simulačního modelu

Cíl kapitoly: Z výše získaných znalostí a poznatků zde bude uveden teoretický návrh modelu řešící problematiku multicastové metody SSM, který bude východiskem pro praktickou část diplomové práce.

5.1 Komunikace zdroj – klient, protokol IGMPv3

Hlavní a nedílnou součástí systému bude komunikace v rámci síťové vrstvy, a to prostřednictvím protokolu IGMPverze3, který podporuje SSM. Bude zde tedy naznačena komunikace v síti, kterou je v realizaci aplikace aplikována a simulována. Následující obrázek zobrazuje, jak vypadá přihlašování, komunikace a odhlašování klientů v síti [5]. Zobrazená skutečnost odpovídá komunikaci 2 klientů, aby byla komunikace čitelná a srozumitelná. V reálném případě spolu komunikují tisíce klientů a mnoho zdrojů.



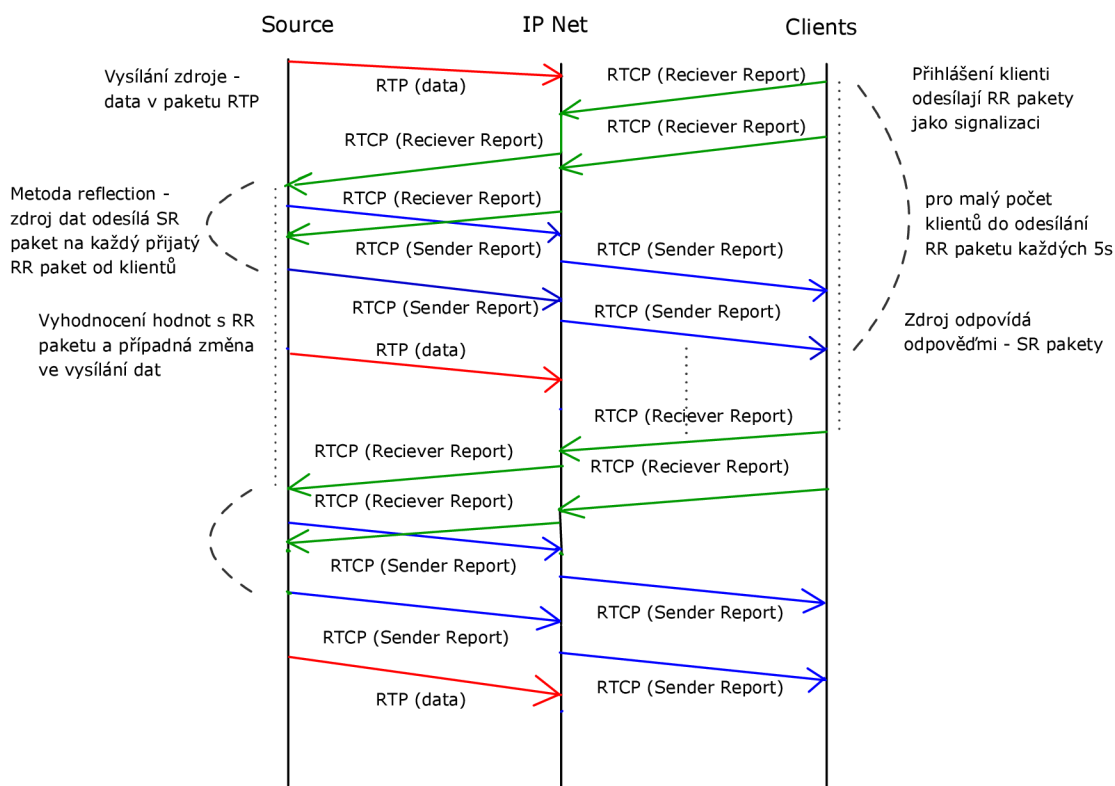
Obr. 17: IGMP zprávy

5.2 Komunikace zdroj – klient, protokol RTP/RTCP

5.2.1 Signalizace – metoda reflection

Dalším a nejdůležitějším úkolem je však uvědomit si, jak vypadá situace, kdy už klienti odeberají data a posílají signalizaci RTCP, a to ve formě Receive Reportu zdroji, který jim následně odpovídá Sender Reportem. Oba pakety SR a RR jsou pakety protokolu RTP/RTCP. Následující obrázek opět ukazuje názorně posílání těchto paketů, opět z důvodů názornosti jenom dvou klientů, a to metodou reflection [13]. Zde už si však zdroj musí hlídat podstatné hodnoty paketu RR, z nichž nejdůležitější jsou ztrátovost

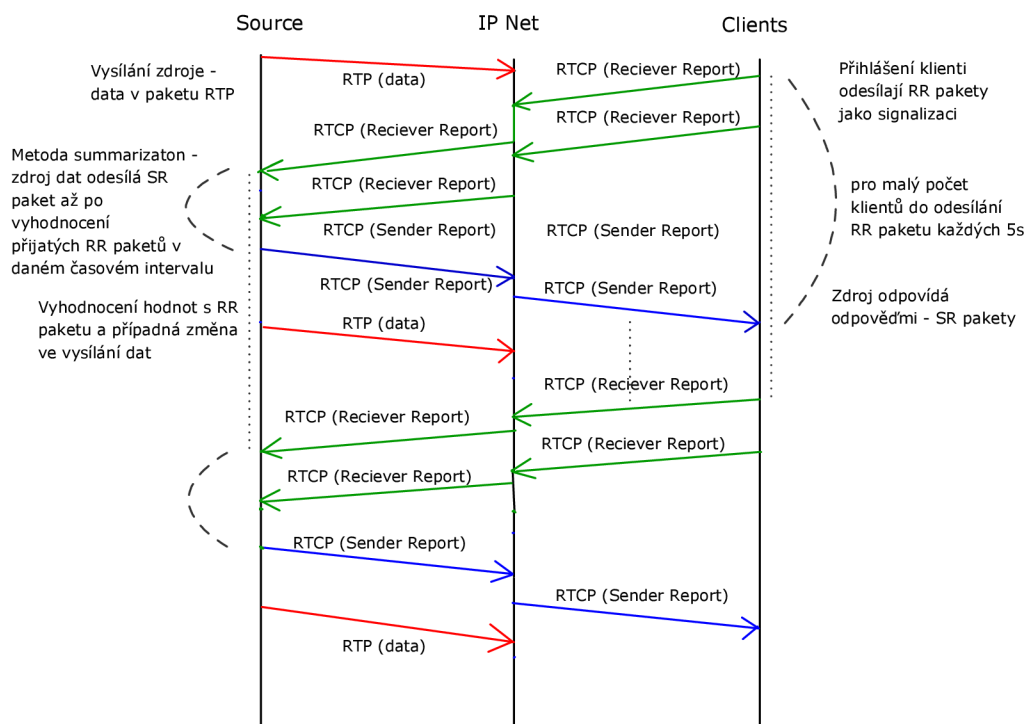
paketů, zpoždění a kolísání zpoždění (jitter). Tyto hodnoty hlídá a při překročení dle potřeby upraví vysílání RTP streamu (dat).



Obr. 18: Zaslání RR a SR paketů - metoda reflection

5.2.2 Signalizace – metoda sumarization

Stejnou skutečnost jako předchozí obrázek zobrazuje i následující, jen s výjimkou použité metody odpovědi ve zdroji dat, který využívá metodu sumarization. Zde stojí za povšimnutí, že zdroj opravdu neodpovídá na každý přijatý RR, ale přijatá data v paketech vyhodnotí, a po uplynutí intervalu přiděleného pro posílání paketu Sender Reportu odešle jediný paket SR. Tato metoda je tedy oproti předchozí významným odlehčením zátěže sítě díky častému posílání těchto zpráv v rámci velkého počtu připojených klientů, řádově se pohybující kolem stovek a více uživatelů. Naopak při menším počtu klientů metoda zatěžuje zdroj v činnosti. Je tedy dobré si uvědomit, jak velká relace klientů bude pro daný zdroj použita a následně zajistit jakou metodu v něm použít.



Obr. 19: Zaslání RR a SR paketů - metoda reflection

5.3 Výpočet intervalů pro odesílání RR a SR paketů - Reflection

Z výše popsaného je tedy zřejmé, jak komunikace probíhá. Proto, než je přikročeno k návrhu a vlastní realizaci, je třeba udělat si přehled časových intervalů pro odesílání SR a RR reportů v závislosti na počtu přihlášených klientů do skupiny. Výpočet bude aplikován na vzorcích (4) pro RR reporty a (5) pro SR reporty. Pro názornost bude zvolen počet klientů na 100, 200, 500, 1000, 10000. Tyto výpočty budou dále zkoumány v závislosti na šířce pásma. Zde byly zvoleny dvě hodnoty a to 0,5Mbps a 1Mbps.

Vzor výpočtu intervalu pro odeslání RR: standardní velikost paketu je 92byte (736b), šířka pásma 1Mbps, počet klientů 1000

$$T_{RR} = \frac{PL_{RR} \times n_T}{0,0375 \times BW_T} = \frac{736 \times 500}{0,0375 \times 1000000} = 9,813s \Rightarrow 10s$$

Vzor výpočtu 1: Výpočet intervalu odeslání paketu Receive Report

Vzor výpočtu intervalu pro odeslání SR: standardní velikost paketu je 60byte (480b), šířka pásma 1Mbps, počet zdrojů 1.

$$T_{SR} = \frac{PL_{SR} \times n_T}{0,0125 \times BW_T} = \frac{480 \times 1}{0,0125 \times 1000000} = 0,0384s \Rightarrow 0,04s$$

Vzor výpočtu 2: Výpočet intervalu odeslání paketu Sender Report

Tab. 1: Hodnoty časových intervalů RR a SR = 0,5Mbps

Šířka pásma[Mbps]	0,500				
Počet klientů [/]	100	200	500	1000	10000
T_{RR} [s]	3,925 **	7,851	19,627	39,253	392,533
T_{SR} [s]	0,0768				

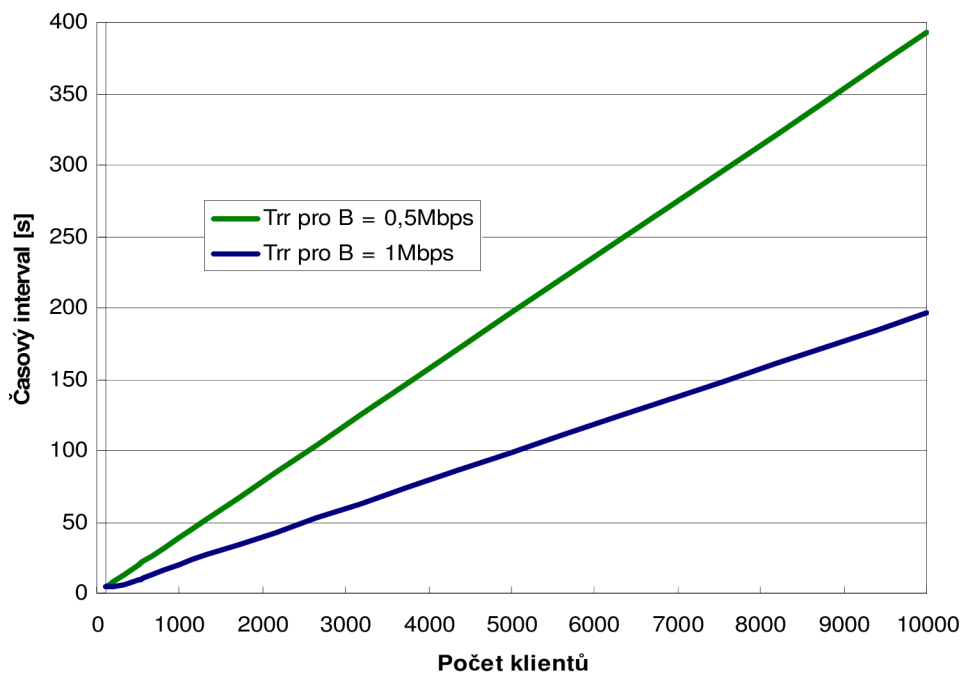
Tab. 2 : Hodnoty časových intervalů RR a SR pro B = 1Mbps

Šířka pásma[Mbps]	1,000				
Počet klientů [/]	100	200	500	1000	10000
T_{RR} [s]	1,962 **	3,925 **	9,813	19,627	196,267
T_{SR} [s]	0,0384				

Pozn. **dle rfc 3550 je minimální hodnota pro odesílání intervalu 5s.

Z vypočtených hodnot je názorně vidět nárůst časových intervalů při zvětšujícím se počtu klientů připojených do multicastové skupiny. Každý účastník má vymezený interval, kdy odesílá svůj RR paket a další odešle až po uplynutí tohoto intervalu. Je tedy na něm, zda-li bude čekat několik desítek či stovek sekund při rozsáhlé relaci, než zdroj pakety vyhodnotí a případně učiní taková opatření, aby bylo požadavkům klientů vyhověno.

Nárůsty hodnot časového intervalu v závislosti na počtu klientů pro dvě zvolené šířky pásma zobrazuje obrázek č. 20.

**Obr. 20: Závislost časových intervalů T_{RR} na počtu klientů**

5.4 Výpočet intervalů pro odesílání RR a SR paketů - Summarization

Při výpočtech časových intervalů odesílání SR paketů metodou summarization (spojování paketů) je třeba si uvědomit, že velikost odesílaného paketu SR, která je značena PL_{RSI} , narůstá s počtem uživatelů, a to tentokrát podle vzorce (7). Velikost intervalu paketu RR je v této metodě stejná jako v metodě reflection. Proto zde dále tyto hodnoty uváděny nebudou. Nejdříve budou vypočítány velikosti těchto paketů v závislosti na počtu uživatelů. Následně jsem zkusil hodnoty aplikovat ve vzorci (5). Pro názornost opět byl zvolen počet klientů na 100, 200, 500, 1000, 10000 jako v předchozím bodě. Tyto výpočty byly následně dále zkoumány pro šířky pásma 0,5Mbps a 1Mbps.

Vzor výpočtu velikosti paketu SR pro metodu summarization: Počet klientů 500, šířka pásma 0,5Mbps, komprese nulová, tzn. $m=1$.

$$PL_{RSI} = 76 + 96 \log_2((n/m) + 1) = 76 + 96 \times \log_2((500/1) + 1) = 936,992 \text{ Byte} \Rightarrow 7464b$$

Vzor výpočtu 3: Výpočet velikosti paketu SR pro metodu summarization

$$T_{RSI} = \frac{PL_{RSI} \times n_T}{0,0125 \times BW_T} = \frac{7464 \times 500}{0,0125 \times 500000} = 599,68s \Rightarrow 10 \text{ min}$$

Vzor výpočtu 4: Výpočet intervalu odesílání paketu SR pro metodu summarization

Z výpočtu a vzorce je názorné, že bez použití tzv. agregace sumarizačních uzlů do stromové struktury je tato metoda pro vyšší počet účastníků naprosto nevyhovující.

Je zde jasně vidět nárůst časových intervalů odesílání jednoho paketu SR metodou summarization, a zvětšující se velikost odeslaných dat v paketu v závislosti na počtu přihlášených klientů. Tyto hodnoty jsou neúměrně velké, proto zde lze jasně pochopit, proč se s výhodou využívá komprese velikosti paketu. Čím je tato komprese vyšší (tzn. čím více je relace rozdělena do stupňů stromu, kde si tyto stupně, známe jako sumarizační uzly, předávají informaci), tím menší bude velikost paketu, a také menší doba časového intervalu pro odeslání paketu SR. S výhodou se tedy využívá této metody. Následující výpočty ukazují, jak by se velikost paketu redukovala v závislosti na kompresi tohoto paketu díky sumarizačním uzlům stromové struktury dané relace.

Tab. 3: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 0,5Mbps, bez komprese dat

Šířka pásma [Mbps]	0,500				
Počet klientů [/]	100	200	500	1000	10000
Velikost paketu SR [b]	5720	6488	7496	8264	10816
T_{RSI} [s]	91,520	207,516	599,680	1322,240	17305,600

Tab. 4: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 1Mbps, bez komprese dat

Šířka pásma [Mbps]	1,000				
Počet klientů [/]	100	200	500	1000	10000
Velikost paketu SR [b]	5720	6488	7496	8264	10816
T_{RSI} [s]	45,760	103,808	299,840	661,120	8652,800

Při výpočtu signalizačního intervalu pro agregaci sumarizačních uzlů do stromové struktury budeme vycházet z uvedeného vztahu (1) a (2). Předtím, než přikročíme k samotným výpočtům, bylo navrženo rozvržení sumarizačních uzlů takto:

Pro 100 klientů bude mít struktura stromu 1 level a 10 sumarizačních uzlů, a každý bude mít pod sebou deset koncových klientů.

Stromová struktura pro 200 a 500 klientů je totožná, pouze konečný počet koncových uživatelů bude pro každý nadřazený sumarizační uzel 20 a 50 klientů. Dá se tedy usoudit, že pokud bude stejná stromová struktura, časové zpoždění bude konstantní, ale zvětšovat se bude na velikosti počtu klientů pod jednotlivým sumarizačním uzlem.

1000 klientů bude uspořádáno ve struktuře, která už bude mít o jeden level ve stromové struktuře více než předchozí hodnoty.

Ta samá pravidla platí pro relace o 10000 klientech, opět se zvýšením levelu stromové struktury agregace o jeden oproti předchozímu počtu 1000.

Vzor výpočtu intervalu pro odeslání RR: standartní velikost paketu je 92byte (736b), šířka pásma 1Mbps, počet klientů 1000:

Nejprve se vypočítá interval odeslání koncových klientů k nadřazenému sumarizačnímu uzlu $RI_{end \rightarrow smr(I-1)}$:

$$RI_{end \rightarrow smr(I-1)} = RI_{RR} = \frac{PS_{RR} \times n_{gend}}{0,75 \times BW_{RTCP}} = \frac{736 \times 10}{0,75 \times 50000} = \underline{0,393s} \quad ,$$

kde PS_{RR} je velikost RR paketu, n_{gend} je počet klientů pod sumarizačním uzlem a BW_{RTCP} je šířka pro RTCP signalizaci z celkové šířky pásma.

Následně se aplikuje vzorec (1) pro výpočet velikosti spojovaného paketu:

$$PS_{RR-RSI} = 608 + 3 \times (96 + 256 \times \log_2(10)) = \underline{2501b}$$

Dále se vypočítá prodleva předání paketu sumarizačního uzlu k jeho nadřazenému uzlu $RI_{smr \rightarrow smr(I-1)}$:

$$RI_{smr \rightarrow smr(I-1)} = RI_{RR-RSI} = \frac{PS_{RR-RSI} \times n_{gmsr(i)}}{0,75 \times BW_{RTCP}} = \frac{2501 \times 10}{0,75 \times 50000} = \underline{0,667s} \quad ,$$

kde $n_{gmsr(i)}$ je počet sumarizačních uzlů patřící nadřazenému sumarizačnímu uzlu.

Tyto vypočítané mezihodnoty je třeba na závěr dosadit do vzorce (2) pro výpočet celkového časového intervalu. Celkový interval pro odeslání reportu v hierarchické agregaci RI_{SS} můžeme vypočítat takto:

$$RI_{SS} = RI_{end \rightarrow smr(I-1)} + \sum_{i=I-1}^{i=1} RI_{smr(i) \rightarrow smr(i-1)} + RI_{smr(0) \rightarrow mem} = 0,393 + (0,667 \times 3) + 5 = \underline{7,394s}$$

Při návrhu aplikace a hlavně metody summarization je proto dobré zvolit kompresní poměr, aby se velikost posílaného paketu redukovala a časový interval pro odeslání byl přijatelný. Následující tabulky ukazují velikosti paketů SR v závislosti na počtu klientů a jejich časové intervaly.

Z vypočtených hodnot je názorně vidět nárůst časových intervalů odesílání jednoho paketu SR metodou summarization a zvětšující se velikost odeslaných dat v paketu v závislosti na počtu přihlášených klientů. Tyto hodnoty jsou neúměrně velké, proto zde lze jasně pochopit, proč se s výhodou využívá komprese velikosti paketu a hierarchické agregace klientů do stromové struktury. Čím je tato komprese vyšší (tzn. čím více je relace rozdělena do stupňů stromu, kde si tyto stupně, známe jako sumarizační uzly, předávají informaci), tím menší bude velikost paketu a také menší doba časového intervalu pro odeslání paketu SR.

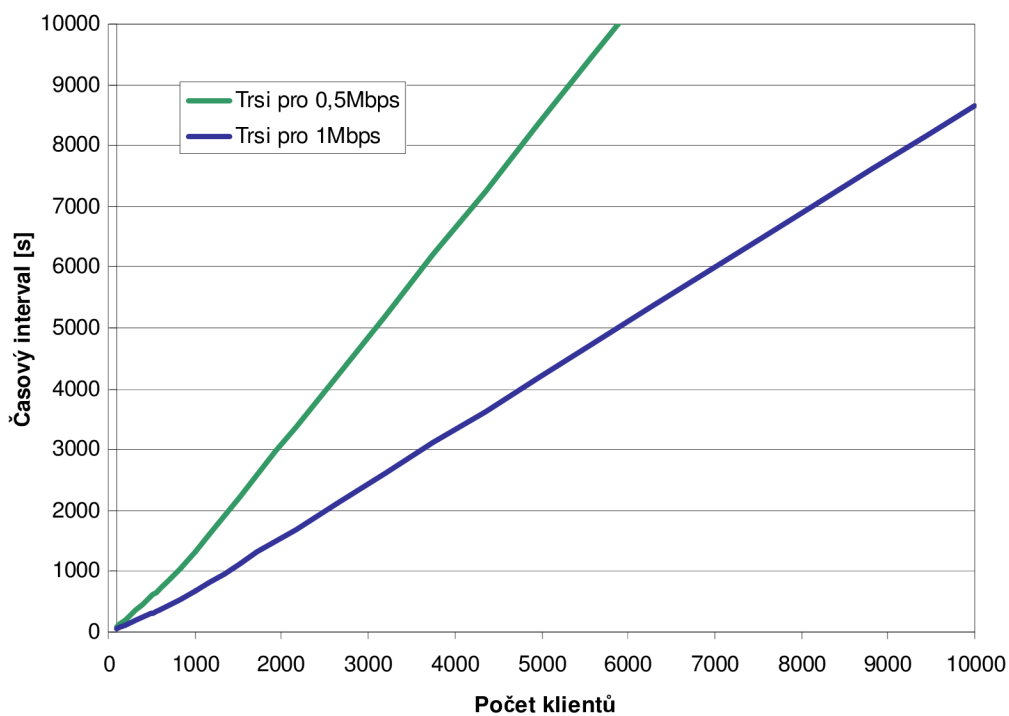
Nárůsty hodnot časového intervalu v závislosti na počtu klientů pro velikosti paketu SR bez kompresního poměru a pro obě šířky pásma zobrazuje následující obrázek. Z něj je naprosto zřejmé, jak je časový interval obrovský, a proto abychom metodu mohli s úspěchem používat, musíme relaci rozdělit do několika vrstev dle obrázku č.16. Je zapotřebí si uvědomit, pro jaký počet klientů bude zdroj data vysílat, jinými slovy, jak tedy bude velká relace, vhodně zvolit počet sumarizačních uzlů, a tím zvolit kompresní poměr. Z toho plyne zmenšení velikosti paketu pro odeslání, a také kratší prodleva odesílání tohoto paketu zpět ke klientům. Viz následující tabulka.

Tab. 5: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 0,5Mbps, s kompresí

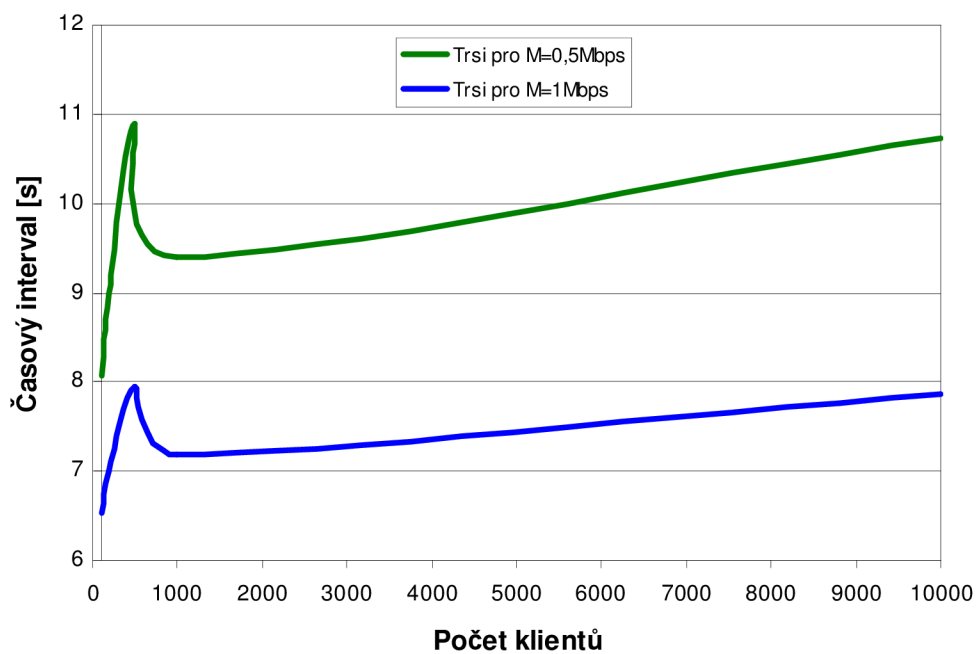
Šířka pásma[Mbps]	0,500				
Počet klientů [/]	100	200	500	1000	10000
Komprese [/]	2	2	2	3	4
Počet klientů pod sumarizačním uzlem	10	20	50	10	10
Velikost paketu SR[b]	2501	3013	3689	2501	2501
T_{RSI} [s]	8,060	8,999	10,897	9,394	10,728

Tab. 6: Hodnoty velikosti paketů a časových intervalů pro odesílání paketu SR metodou summarization pro šířku pásma 1Mbps, s kompresí

Šířka pásma[Mbps]	1,000				
Počet klientů [/]	100	200	500	1000	10000
Komprese [/]	2	2	2	3	4
Počet klientů pod sumarizačním uzlem	10	20	50	10	10
Velikost paketu SR[b]	2501	3013	3689	2501	2501
T_{RSI} [s]	6,530	6,999	7,949	7,197	7,864

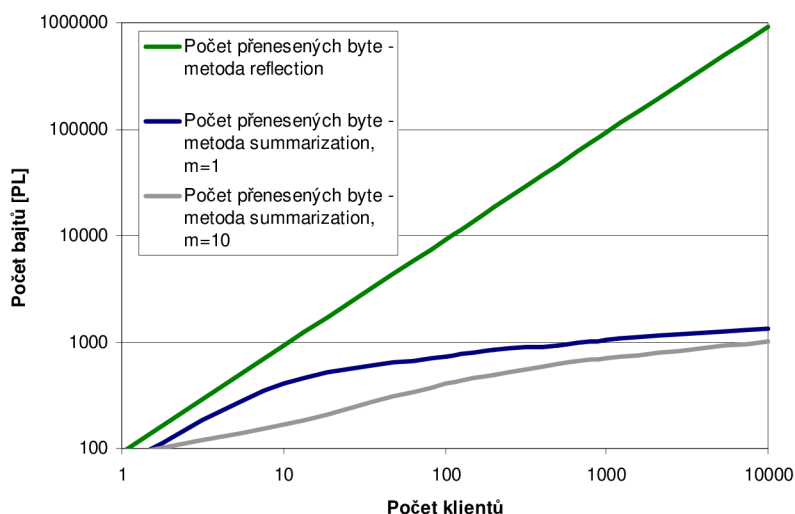


Obr. 21: Závislost časového intervalu T_{RSI} na počtu klientů metody summarization bez kompresního poměru dat



Obr. 22: Závislost časového intervalu T_{RSI} na počtu klientů metody summarization v hierarchické agregaci a s kompresí dat

Názorným příkladem je, jaká bude zátěž linek při signalizaci, tedy kolik Byte bude za použitých metod po linkách posíláno. Využitím vzorců (5) a (6) je vypočtena závislost počtu přenesených byte na počtu připojených klientů. Zde jsou tedy jasně zobrazeny rozdíly obou používaných metod reflection (přeposílání paketů) a summarization (spojování paketů ve zdroji) v závislosti na počtu přenesených signalizačních dat. Srovnáním obou metod dle vypočtených hodnot můžeme tvrdit, že metoda reflection je časově flexibilnější a rychlejší, ale při větším počtu příjemců bude zahlcovat síť velkým objemem odesílaných dat. Oproti tomu metoda summarization (spojování paketů) by měla vytvářet zpoždění, ale za to by měla být efektivnější, co se týče potřebné šířky pásma.



Obr. 23: Porovnání počtu přenesených byte (vyžadované šířky pásma) v jednom signalizačním intervalu pro metody reflection a summarization

5.5 Podstatné hodnoty signalizace RTCP

Při samotné realizaci modelu, který simuluje komunikaci klienti-zdroj metodou SSM, se zabývá reakcí zdroje na problémy spojené s již zmíněnými nejdůležitějšími vlastnostmi vysílání RTP dat, což jsou ztrátovost, zpoždění a jitter. Je vhodné si zde uvědomit, že pokud chce zdroj zajistit opravdu kvalitní vysílání (stream), například IPTV, měly by hodnoty těchto veličin být pro:

- Ztrátovost paketů – řádově desetiny %
- Zpoždění - řádově několik stovek ms
- Kolísání zpoždění – řádově několik desítek ms

Tyto hodnoty ale jsou pro každého zprostředkovatele vysílání subjektivní, každý zdroj má jiné hranice, kdy a za jakých hodnot těchto veličin upravuje své vysílání.

6. Návrh Aplikace simulující multicastové spojení

Cíl kapitoly: V této kapitole je uvedena problematika jazyka SDL, vývojového prostředí Telelogic Tau a komunikace v navrženém systému, založeném na přenosu dat pomocí technologie SSM.

6.1 Úvod do SDL

6.1.1 Historie jazyka SDL

Jazyk SDL je standardním jazykem určeným ke specifikaci popisu systému. Byl vyvinut a normalizován organizací ITU (International Telecommunication Union). Jazyk SDL se začal vyvíjet v roce 1972. První verze byla dokončena v roce 1976, další pak v letech 1980, 1984 a 1988. Poslední dvě verze jazyka jsou podstatně rozšířeny díky jeho dlouhodobým studiím a používáním. Další rozšíření jeho vlastností bylo v roce 1992, a to na objektově orientovaný jazyk SDL. V roce 1992 se ze standardu SDL osamostatnil jazyk MSC (Message Sequence Charts). Později na tomto základě vznikly sekvenční diagramy UML (Unified Modelling Language). Standardy SDL a UML se spojují v společném cíli, kterým je UML 2.0: jeden společný jazyk pro vizuální softwarové inženýrství, spojující výhody obou dosavadních standardů. Standard UML 2.0 byl specifikován v roce 2002. UML 2.0 přebírá některé vlastnosti jazyka SDL, například modelování architektur pomocí komponent s definovaným rozhraním či modelování chování pomocí stavových automatů.

6.1.2 Oblast aplikací a výhody jazyka SDL

V současné době je jazyk SDL využíván hlavně v oblasti telekomunikací. Jsou jím popsána některá doporučení ITU, spojovací systémy 4. generace, terminály ISDN (Integrated Service Digital Network), atd. Sloužil i pro vývoj československého spojovacího systému EDISS (Elektronický Digitální Spojovací Systém) a popisu jeho funkcí. Jazyk SDL je používán pro systémy, které pracují v reálném čase, kde řízení systému je distribuováno či rozděleno mezi několik procesů, které spolu navzájem komunikují díky vzájemné výměně zpráv.

Hlavními výhodami jazyka SDL jsou:

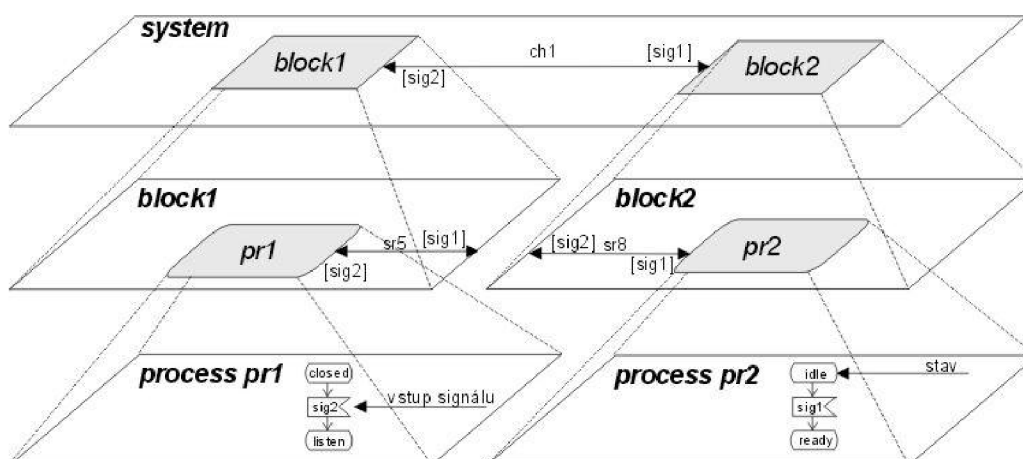
- schopnost jazyka pokrýt návrh systému od požadavku až po implementaci
- prezentace v grafické formě
- poskytuje informaci jak o dynamickém chování, tak i struktuře systému
- umožňuje zachytit úroveň abstrakce chování systému v závislosti na požadované hloubce detailu
- objektově orientovaný popis komponent SDL
- modelový základ komunikačních procesů (rozšířené automaty s konečným počtem stavů)

6.1.3 Formy prezentace

Přednost jazyka SDL spočívá v jeho grafické verzi SDL/GR (Graphical Representation), která dělá specifikaci srozumitelnou a přehlednou. Grafická syntaxe je doplněna textovou syntaxí, protože ne vše lze nebo je vhodné vyjádřit pomocí grafických symbolů, např. abstraktní datový typ. Později vznikla také textová verze jazyka SDL/PR (Phrase Representation), která využívá jenom textovou syntaxi a vychází z tzv. "abstraktní gramatiky".

6.1.4 Základní vlastnosti SDL

Jazyk SDL je organizován hierarchickou úrovní. Na nejvyšší úrovni je definován systém, který tvoří libovolný počet bloků. Každý blok pak obsahuje procesy, které tvoří hlavní strukturu navrhovaného programu. Jazyk je založen na popisu modelovaného systému pomocí paralelně běžících konečných stavových automatů (KA) rozšířených o řadu užitečných vlastností. Základními prvky KA jsou stavy a přechody mezi nimi. Jeden stav je počáteční (start), a v tomto stavu začíná činnost konečného automatu. Na základě splnění určité podmínky (např. přijetí daného signálu) může přejít KA z jednoho stavu do druhého. Náznornější představu o hierarchickém upořádání struktury ukazuje obr.24[1]



Obr. 24: Strukturované rozdělení systému

6.1.5 Statický popis jazyka SDL

6.1.5.1 Systém

Systém tvoří vrchol úrovně abstrakce (tvoří kořen popisu) a reprezentuje statickou strukturu. Systém je to, co SDL popis specifikuje: abstraktní stroj, komunikující se svým okolím. Každý systém musí být pojmenovaný. Systém obsahuje jeden nebo více bloků, které jsou spolu vzájemně spojené nebo spojené s okolím systému pomocí kanálů. Kanály jsou hlavním prostředkem přenášení signálů (informací) mezi bloky navzájem nebo mezi bloky a okolím. Popis systému se v SDL/GR nazývá diagram systému.

6.1.5.2 Blok

Blok je část systému, která může sloužit k různým účelům (vývoj, popis, vysvětlování funkcí atd.) jako samostatný objekt. Signálové cesty spojují procesy uvnitř bloku navzájem nebo s okolím bloku. Aby měla existence bloku smysl, tak musí obsahovat alespoň jeden typ procesu. Popis bloku navazuje na popis systému, tedy signály vstupující do bloku pomocí kanálu musí být shodné se signály přenášenými signálovými cestami na procesy napojené na tento kanál.

6.1.5.3 Proces

Proces v jazyku SDL se bere jako tzv. rozšířený konečný automat (má konečný počet stavů) a má vlastní pomocnou paměť, kde může uchovávat své proměnné. Proces může obsahovat i vnořené procesy. Nejvíce vnořený proces popisuje chování (svým stavovým automatem). Vnoření agentů procesu nepracují zároveň, ale střídavě, že v jednom okamžiku pracuje nejvýše jeden vnořený proces.

6.1.5.4 Procedura

Procedura, úroveň jazyka, která dovoluje rozdělení procesů do několika úrovní abstrakce a umožňuje opakování položek. Definice procedur mohou být obsaženy pouze v definici procesu, v definici služeb nebo v definici procedur. Procedura je viditelná jen na té úrovni, na které je definována. Procedura může být konečným automatem uvnitř procesu a je vytvořena, je-li proveden příkaz volání procedury (call) a je ukončena, je-li proveden příkaz návrat z procedury (return). Pro proceduru platí stejná množina platných vstupních signálů (signalset), jako pro proces, který proceduru vyvolal.

6.1.6 Popis dynamického chování jazyka SDL

Základním prvkem popisu dynamického chování systému v jazyce SDL je proces, jež pracuje nezávisle současně s dalšími procesy. Proto i chování celého systému je založeno na kombinaci chování množství těchto procesů. Popis procesu definuje typ procesu. V systému může současně existovat několik procesů (instancí), daných popisem jednoho typu procesu. Proces daného typu může vzniknout buď při vzniku systému, nebo dynamicky v době existence systému pomocí některého již existujícího procesu, který se v tomto případě nazývá rodič. Množství procesů téhož typu (instancí) je určeno dvěma čísly, které jsou uvedeny za jménem procesu. První číslo uvedené po jménu procesu, označuje množství procesů, které vzniknou současně se vznikem systému. Druhé číslo označuje maximální množství procesů daného typu, které může současně existovat v systému. Jestliže druhé číslo je vynecháno, pak smí v systému existovat jakékoliv množství procesů téhož typu. Každý proces má jedinečný identifikátor procesu PID (Process Identifier) v rámci celého systému. PID je také speciální datový typ jazyka.

Spolupráce mezi procesy v jazyku SDL je asynchronní, a to prostřednictvím diskretních zpráv zvaných signál. Ty může proces vysílat do svého okolí nebo také přijímat ze svého okolí. Proto se i okolí procesu musí chovat způsobem definovatelným v jazyku SDL. Chování procesu je deterministické a jeho reakce na vnější popudy (ve formě signálů) musí být v souladu s jeho popisem. Signál je vždy doplněn o identifikátor vysílajícího a přijímacího procesu jako dodatek k přenášené hodnotě signálu. Tím přijímací proces zná

adresu vysílajícího procesu. Tato metoda se potom s výhodou využívá při posílání a adresaci signálů mezi procesy.

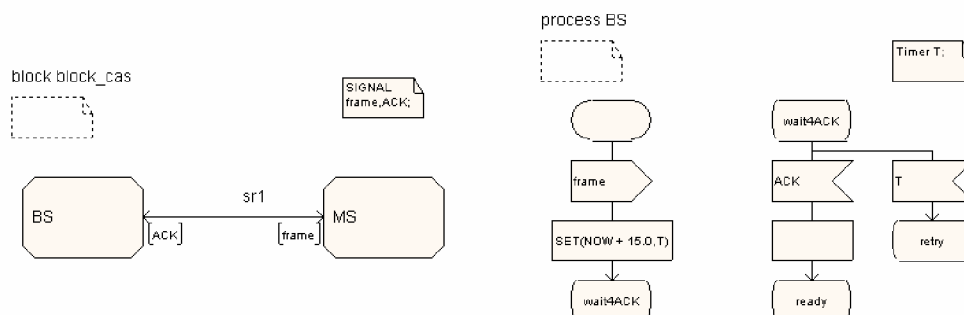
Proces má neomezenou vstupní frontu, do které jsou zařazovány příchozí signály. Proces je buď ve stavu (čeká na popud) nebo provádí přechod mezi dvěma stavy. Přechod je vyvolán vstupem prvního signálu ve vstupní frontě procesu. Jinak řečeno, tento předaný signál je procesem zkonsumován. Během přechodu je možno provádět změny hodnot proměnných, lze využít různé konstrukce rozhodování, vytvořit nové procesy, vyslat signály (na další proces nebo na sebe) a podobně.

6.1.7 Komunikace mezi procesy

V SDL se musí signály vysílat asynchronně, to tedy znamená, že vysílající proces pokračuje bez čekání na potvrzení z přijímacího procesu. Každý proces nebo jeho instance má neomezenou vstupní frontu signálů s režimem FIFO (First In First Out). Přejde-li na proces jakýkoliv signál, jež je obsažen v celé množině platných vstupních signálů (množina je uváděna za klíčovým slovem signalset v symbolu textu umístěném v grafu procesu), je zařazen do vstupní fronty příslušné instance procesu. Úplná množina platných vstupních signálů tak definuje ty signály, které může proces přijmout. Jestliže v daném stavu není vstupní fronta prázdná, první signál (ve frontě FIFO) je odstraněn z vstupní fronty (spotřebován), tj. předán na proces. Je při tom provedena kontrola, jestli tento signál může vyvolat přechod. Jestliže ano, provede se přechod, jestliže ne, signál je zrušen.

6.1.8 Čas v jazyce SDL

Je-li popisováno chování systému, je vhodné v některých případech určit časové omezení. Za tímto účelem je v jazyce SDL definována konstrukce časovače. Časovač je objekt definovaný na úrovni procesu, který je schopný generovat signál od časovače a tento signál vložit do fronty. Časovač se definuje klíčovým slovem Timer, za kterým následuje jméno časovače. Pro specifikaci doby aktivity časovače je použit výraz NOW (předdefinovaný v sortě Time, která je podobná sortě Real). Konstrukce NOW vždy přiřazuje skutečný čas během svého použití v popisu systému.



Obr. 25: Deklarace a nastavení časovače

Názornou ukázkou funkce časovače znázorňuje obrázek výše, popisující princip komunikace mezi základnovou a mobilní stanicí. Blok `Block_cas` obsahuje dva procesy, `BS` (Base Station) vysílající signál `frame` (rámec) a `MS` (Mobile Station) vysílající signál potvrzení `ACK` (Acknowledge). Po odeslání signálu `frame` proces `BS` nastaví časovač na hodnotu 15 jednotek (jednotka v jazyku SDL může být nastavena na `sec`, `msec`, `psec` aj.) a čeká na příjem signálu `ACK` od procesu `MS`. Po vypršení této doby (proces `MS` nevyslal potvrzení) přechází proces `BS` do stavu `Retry`, kdy se snaží o nové spojení. [1]

6.1.9 Adresování v jazyce SDL

Jakýkoliv signál vyslaný procesem musí mít svého příjemce. Cílový proces může být specifikován dvěma způsoby:

- ▶ *explicitně* (výslovně) zapsanou adresou v konstrukci výstupu.
- ▶ *implicitně* (vnitřně) vynecháním adresy v konstrukci výstupu.

V obou případech však musí platit, že existuje jen jedna komunikační cesta ve formě signálových cest a kanálů mezi vysílajícím a přijímajícím procesem. V případě více možných cest lze dovolené komunikační cesty pro signál definovat v konstrukci výstupu, kde jsou výslovně určeny buď signálové cesty, nebo kanály, kterými je signál přenášen. Nelze použít kanály a signálové cesty současně. Cílový proces musí vždy existovat v době vyslání signálu. Jestliže tomu tak není, vznikne dynamická chyba a další chování systému je nedefinováno.

Každý proces má svoji jedinečnou adresu. Tato adresa není určena uživatelem, ale je vytvořena abstraktním SDL strojem v době vytváření procesu. Tento SDL stroj tak zaručuje, že každý proces má jedinečnou adresu. Adresa procesu není identická se jménem procesu, protože může existovat mnoho instancí daného typu procesu, které mají stejné jméno. Adresy jsou typu identifikátor procesu `Pid` (process identifier), což je dříve zmíněný předdefinovaný datový typ v SDL.

Pro každý proces existují 4 předdefinované výrazy (proměnné) typu `Pid` :

- self* - výraz udávající identifikátor procesu samého. Proces může zjistit v případě potřeby svoji identitu pomocí příkazu `A: = self`, kde proměnná musí být deklarována jako typ `Pid`.
- sender* - výraz v sobě uchovává identifikátor procesu, od kterého přijímající proces naposledy přijal (spotřeboval) signál. To znamená, že hodnota této proměnné je po každém přijetí signálu přepsána. Velmi často se používá pro zjištění adresy procesu, kterému se má potvrdit právě přijatá zpráva.
- offspring* - výraz udávající identifikátor posledně vytvořeného procesu - potomka. Hodnota je vždy přepsána poté, když proces-rodíč vytvoří novou instanci procesu.
- parent* - výraz udávající identifikátor procesu-rodíče, který tuto instanci procesu vytvořil. Hodnota je nastavena pouze jednou, a to v době vzniku procesu. [2]

6.2 Realizace aplikace

Pro návrh modelu jsem použil nejrozšířenější SDL CASE (Computer Aided Software Engineering) nástroj na trhu, což je program Telelogic Tau verze 4.3 švedské firmy Telelogic. Telelogic Tau obsahuje editory MSC, UML a SDL, dále simulátor, validátor a konfigurovatelný generátor výsledného kódu. Telelogic Tau existuje ve verzi pro Windows, Solaris a HP Unix.

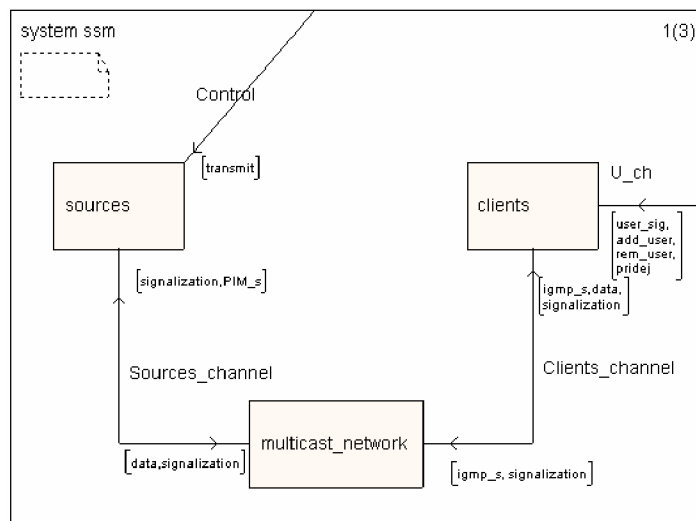
6.2.1 Popis simulačního modelu

Aplikace, neboli simulační model je navržen tak, aby simuloval komunikaci klient – server, a to za použití multicastové metody SSM. Model je schopen simulovat komunikaci prostřednictvím protokolu IGMPv3, tak i protokolu RTP/RTCP a k jeho funkčnosti je třeba ho uživatelsky řídit. Toho je zapotřebí ovšem jenom jednou při počátečním nastavení pravidel relace, množství připojených klientů apod. Dále už všechna další nastavení provádí „jakoby uživatel“, řekněme pomocí například určitého IPTV prohlížeče. Toto ovládání už se děje na jakoby aplikační vrstvě modelu. Model byl navržen, aby pomocí vybraných vrstev OSI/ISO rozlišoval komunikaci. Z tohoto důvodu jsou dílčí části bloku *Clients* rozděleny na patřičné vrstvy, kterým daná komunikace a podporující protokoly náleží. Lze tedy říci, že komunikaci založenou na protokolu IGMPv3 zpracovává proces *Network_layer* a proces *1000_Clients*, což odpovídá síťové vrstvě. A podobně výměnou zpráv a komunikací na bázi protokolů RTP/RTCP se zabývá proces *Application_layer* a proces *1000_RTCP*, jakoby patřící do aplikační vrstvy. Druhý objekt zdroje dat, blok *Source*, již takto rozdělený není, neboť jeho úkolem není žádná komunikace, ale pouhé vysílání paketů RTP a RTCP, tudíž, se dá říci, že pracuje jenom na aplikační vrstvě, řekněme v podobě určitého IPTV serveru. Ten se skládá z části pro samotné vysílání dat do sítě, a dále ze dvou částí pro signalizaci. Ty je také třeba na začátku simulace nastavit, a to podle toho, jestliže budeme chtít simulovat metodu reflection nebo summarization. Třetí a poslední částí modelu je blok *multicas_network*, který má za úkol jak řízení sítě (registrace, odregistrace součástí), tak i vytvoření multicastové skupiny a pro SSM nezbytného SG kanálu.

Díky tomuto modelu můžeme díky MSC diagramům pozorovat jak probíhající komunikaci mezi těmito třemi částmi modelu a jejími procesy, tak ale také simulovat chování spojení zdroj - klienti v závislosti na hodnotách daného spojení a na hodnotách RTCP paketů, které je možné načítat z externích souborů a jejich výsledky také ukládat modelem do jiných externích souborů.

6.2.2 Součásti systému aplikace – blokové schéma

Strukturu aplikace tvoří bloky *sources*, blok *multicast_network* a blok *clients*. Vše zobrazuje následující obrázek. Na něm je zřejmé, jaké hlavní signály si dané bloky mezi sebou vyměňují. Tyto signály ovšem mají definovanou strukturu tak, aby odpovídaly normám daných použitých protokolů. Tato skutečnost bude vysvětlena níže.

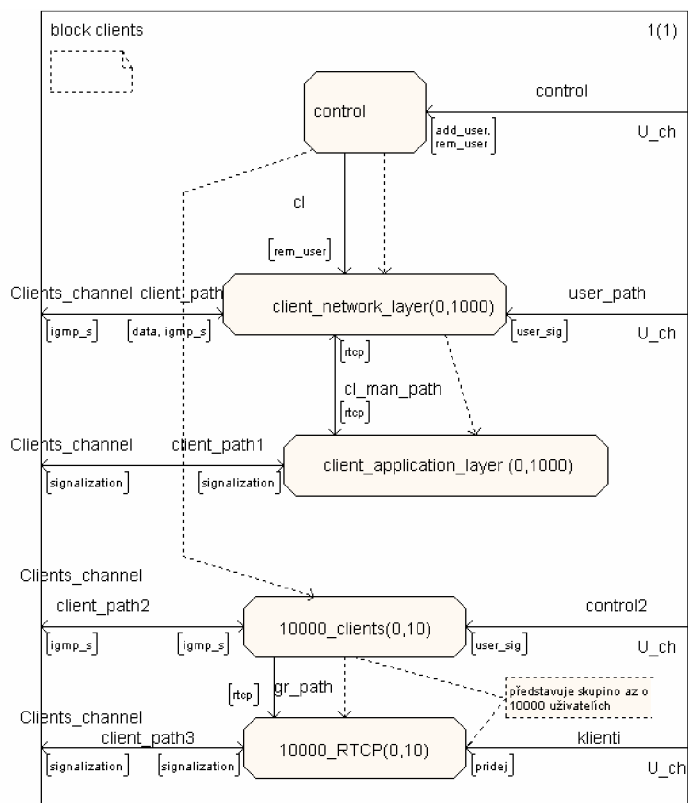


Obr. 26: Blokové schéma simulačního modelu v jazyce SDL

Do bloku CLIENTS jsou od uživatele posílány zprávy pro počáteční nastavení. Jedná se o zprávy *user_sig*, *add_user*, *rem_user*, *pridej*. Tyto signály, jak je zřejmé z jejich názvu, slouží k nastavení počtu klientů, jejich zvyšování či odebrání z modelu aplikace. Podobnou úlohu má také signál *transmit*, jež přikazuje zdroji dat, aby se připravil k vysílání RTP paketů. Následně už probíhá komunikace mezi zobrazenými částmi za pomoci signálů *igmp_s*, *signalization*, *data* a *PIM_s*. Signály *igmp_s* náleží protokolu IGMPv3 a jedná se o žádosti, dotazy a odpovědi protokolu, které jsou JOIN, QUERY, REPORT, LEAVE a mají pevně danou strukturu, která je dána v kapitole 3.1.3. Signál *signalization*, který patří protokolu RTCP, zobrazuje vysílání reportů Sender Report a Receiver Reportů, a jeho struktura je také stanovena podle kapitoly 3.3.2.1 a 3.3.2.2. Pro signál *data* je zvolena struktura paketu RTP s vlastnostmi a informacemi o zdroji dat. Poslední je signál *PIM_s*, který se posílá mezi blokem sítě a zdrojem dat, ovšem nemá pevně danou strukturu. Jeho prací je jen posílat žádosti o začátku či ukončení vysílání dat.

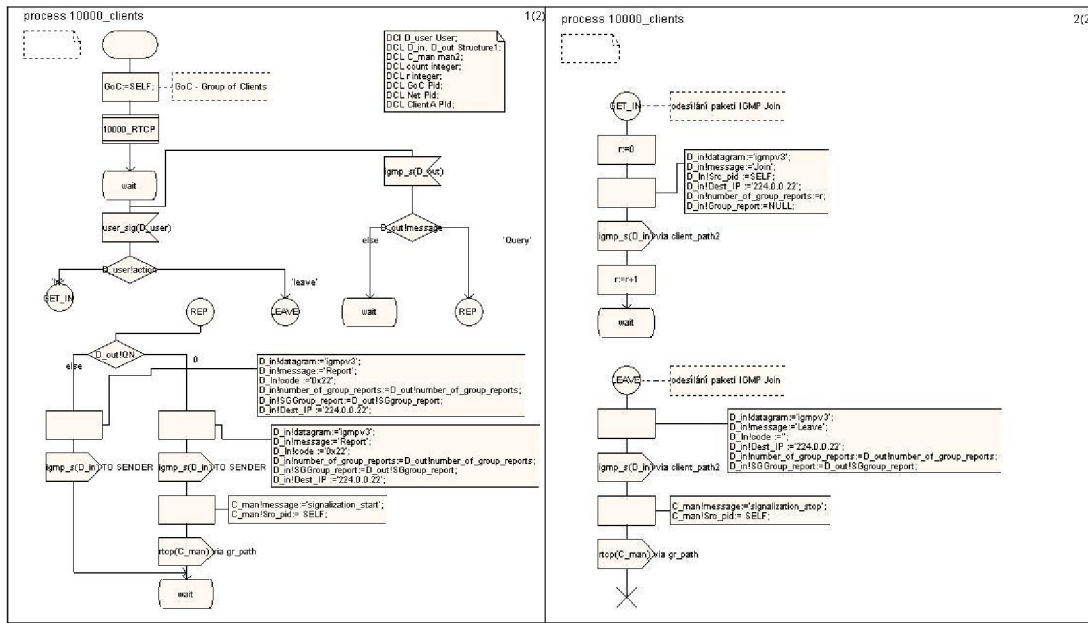
6.2.2.1 Blok clients

Blok clients je aktivní částí tohoto systému. Díky němu se dá ovládat spuštěná simulace a nastavovat její budoucí parametry. Blok tvoří několik procesů, jejichž funkce zde bude vysvětlena. Strukturu bloku zobrazuje obrázek níže.

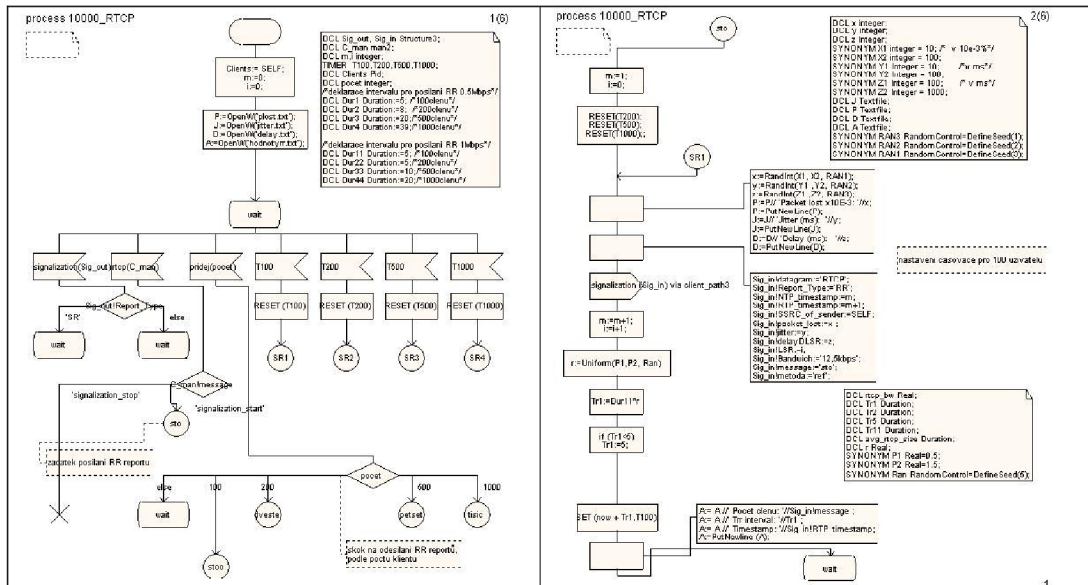


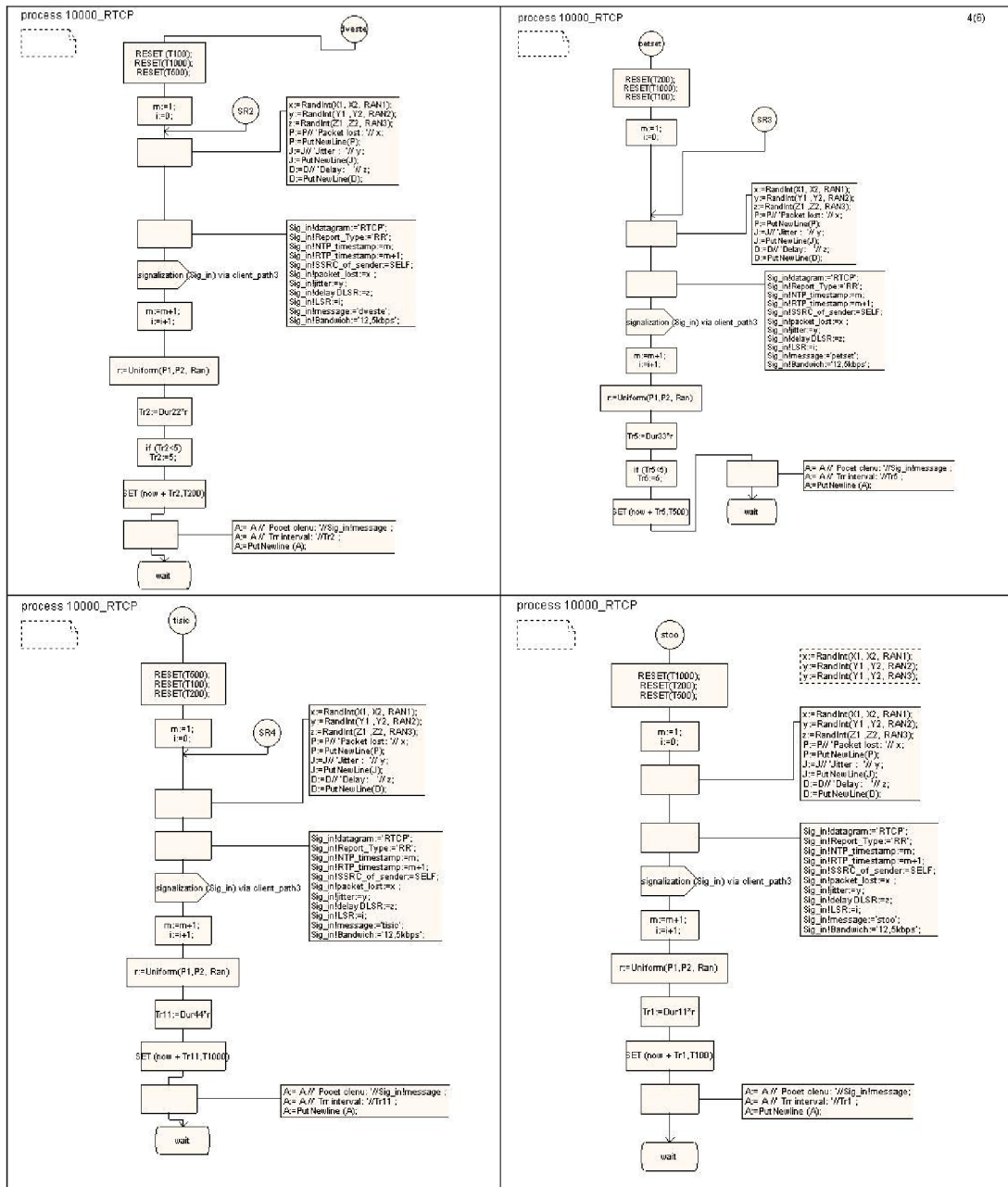
Obr. 27: Procesy tvořící blok clients

Následující procesy jsou samotným jádrem bloku a jeho správné funkce. Ve výčtu procesu tedy začnu od začátku. Prvním procesem je proces *control*, jež slouží jenom k vytvoření instancí procesů pracujících na síťové vrstvě, a to již zmíněných *client_network_layer* a *1000_clients*. Takto vytvořené procesy následně vytvoří své párové procesy pracující na vyšší aplikační vrstvě. Jsou to : *client_aplication_layer* pro *client_network_layer* a *1000_RTCP* pro *1000_clients*. Tyto dvojice jsou prakticky stejné, s tím rozdílem, že spodní dvojice je přednastavena pro vyšší počet klientů, zatímco horní simuluje komunikaci v síti při malém počtu účastníků. Na následujících obrázcích tedy budou ukázány procesy řešící tuto komunikaci v rámci bloku clients.



Obr. 28: Ukázka procesu řešícího komunikaci protokolu IGMPv3 v bloku clients

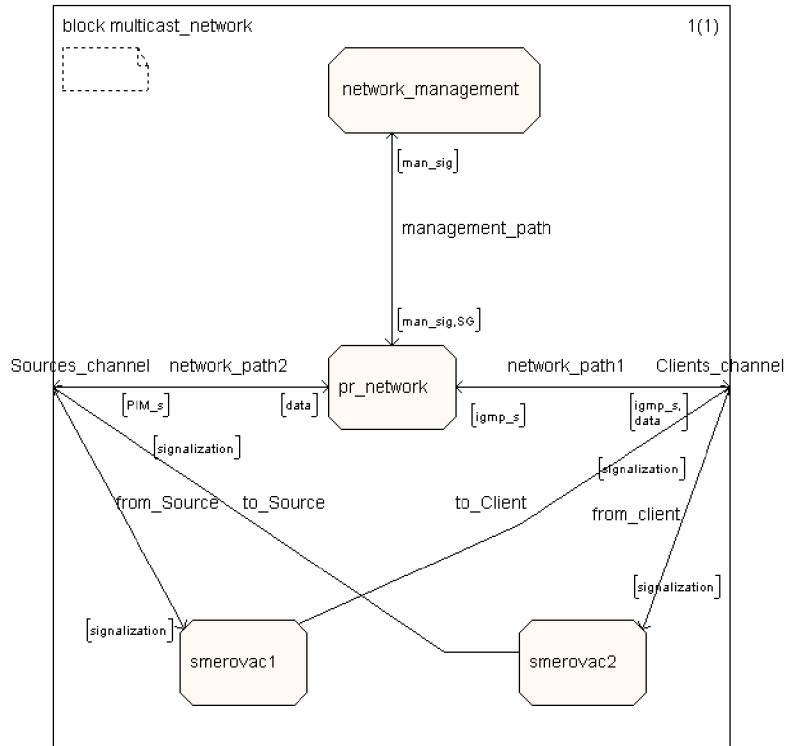




Obr. 29: Ukázka procesu pro posílání reportů v bloku clients

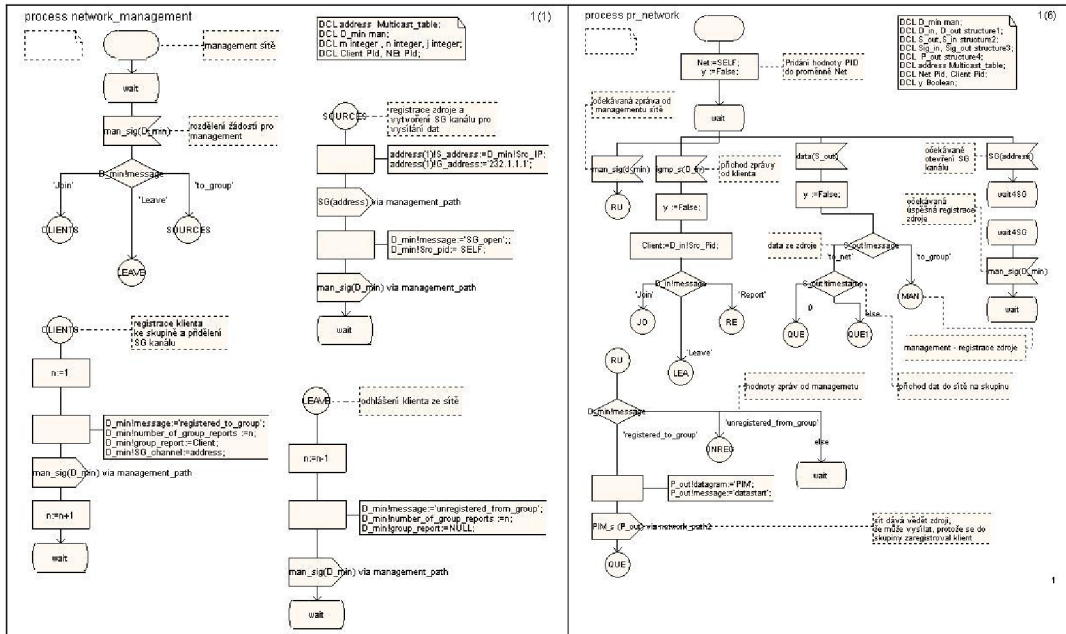
6.2.2.2 Blok multicast_network

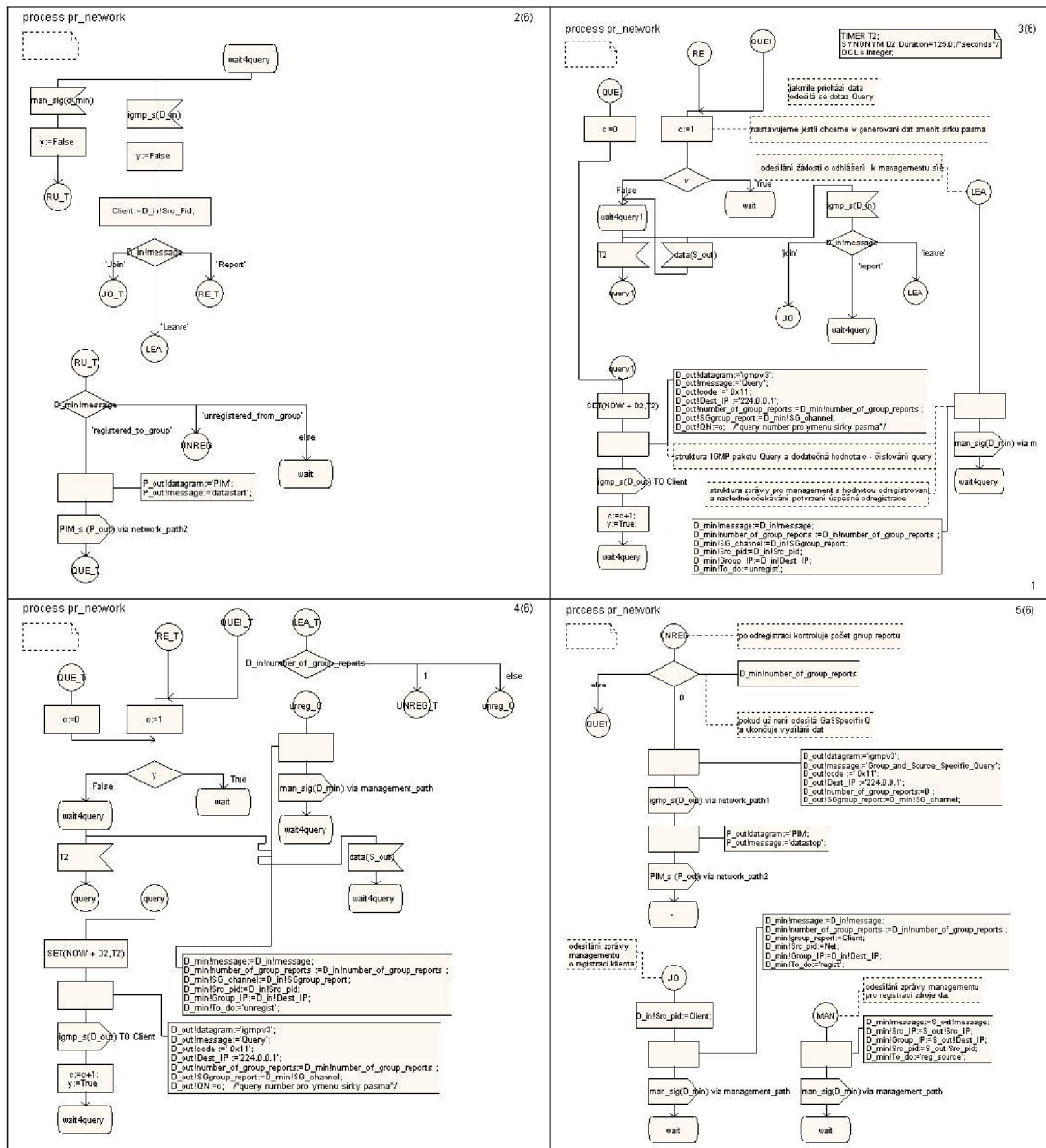
Tento blok je nejdůležitější součástí modelu. Je to hlavní prvek veškeré komunikace, který zpracovává signály jak od klientů, tak od zdroje. Je tvořen procesy *router1* a *router2*, které jenom směřují signalizaci. Hlavní mozek bloku je proces *network_management*, který řeší registrování a odregistrování klientů do skupiny, vytvoření SG kanálu a další problémy spojené s managementem sítě. Druhým důležitým procesem je proces *pr_network*, který působí jako konzument přichozích vstupních dat a dle potřeby je předává managementu. Situaci rozebírá obrázek níže.



Obr. 30: Procesy tvořící blok multicast_network

Strukturu jednotlivých procesů a jejich ukázkou budou zobrazovat následující obrázky, které přiblíží funkci prvků a stavbu takového procesu.



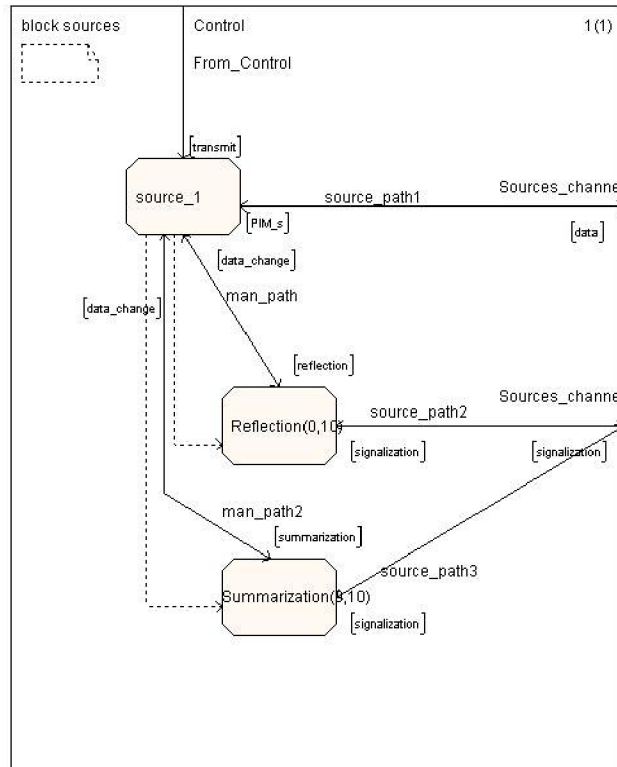


Obr. 31: Ukázka struktury bloku *network_management* a bloku *network*

Struktura procesů obou směrovačů zde není uvedena, neboť není složitá a stará se jen o další směrování signalizačních paketů mezi klienty a zdrojem dat.

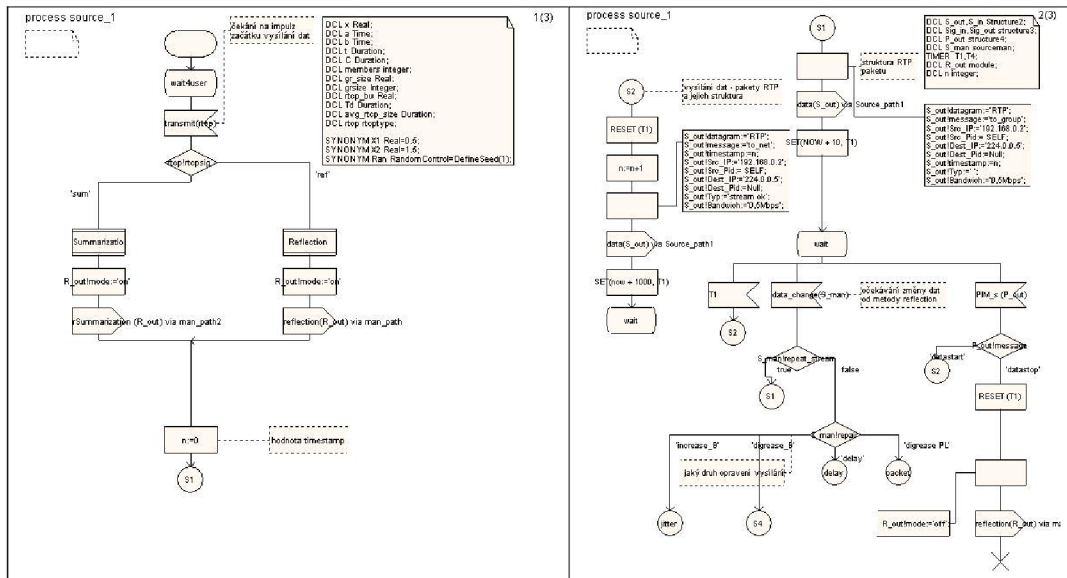
6.2.2.3 Blok sources

Blok *sources* tohoto modelu se skládá ze třech částí, a to z hlavního procesu pro vysílání RTP paketů a dvou zbývajících procedur, které značí použitou metodu RTCP feedbacku. To znamená použít metodu *Reflection* nebo metodu *Summarization*. Samostatně tedy pracují vždycky jenom dvě ze tří částí bloku. Procedury signalizace komunikují s vysílací částí zdroje díky kanálům pro management a zprávám *data_change*, viz. následující obrázky celého bloku a následně i samostatného procesu a vytvořených procedur.



Obr. 32: Procesy tvořící blok sources

Struktura procedur *Reflection* a *Summarization* je naprosto totožná, pouze s rozdílem použitých hodnot časovačů pro odesílání SR reportů. Proto zde bude uvedena jenom část struktury procedury *Reflection* a struktura procesu *source_1*.

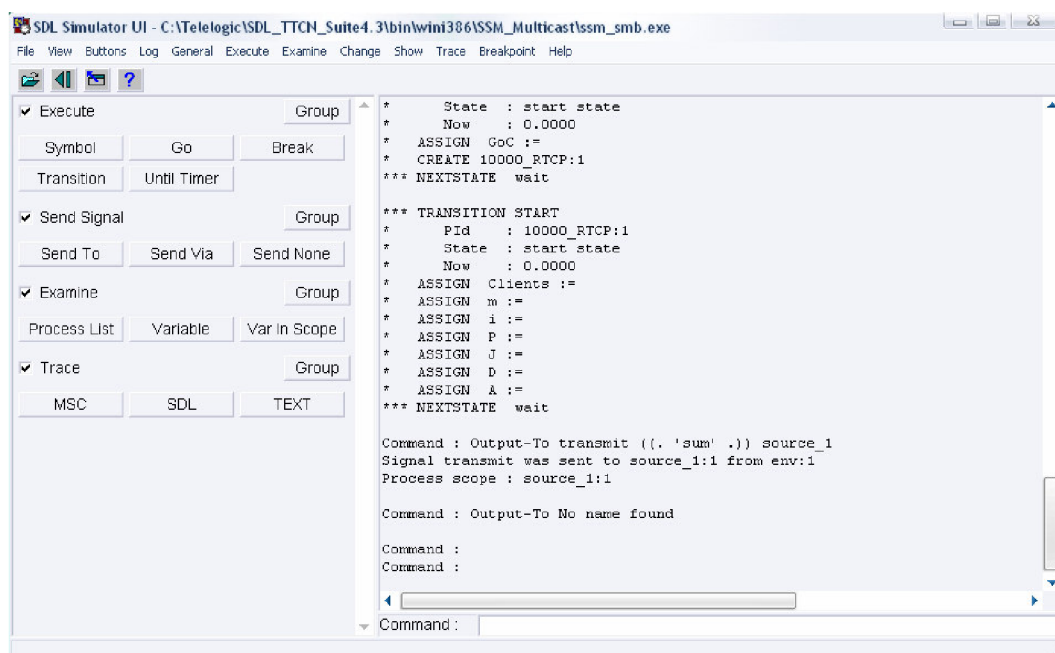


Proto, abychom spustili simulaci vysílání dat na multicastovou skupinu, je třeba uvést zdroj dat do chodu, nastavit signalizační metodu (reflection vs. summarization) zvolit patřičný počet klientů, který lze dále také uměle zvyšovat či snižovat. To provedeme pomocí tlačítka „SENDTO“ a vybereme signál *add_user*. Do pole hodnot vepíšeme hodnotu 1, 2, 3 atd. či 10000, pro simulaci velkého počtu klientů, potvrdíme a signál pošleme na blok *control*, který vytvoří instanci procesu. Potvrdíme opětovným spuštěním tlačítka „GO“.

Podobným nastavením nastavíme zdroj vysílání. Pomocí tlačítka „SENDTO“ zvolíme signál *transmit* spolu s hodnotou „ref“ nebo „sum“ pro zvolení metody a odešleme na blok *source_1*.

V tuto chvíli máme nastaven počet klientů a zdroj vysílání dat. Posledním krokem tedy bude odeslat signál *user_sig*, že klienti chtějí vstoupit do multicastové skupiny a připojit se k vysílání. Opět zvolíme stejný postup. Tlačítko „SENDTO“, signál *user_sig* s hodnotou „in“.

Pozn. Pokud chceme jednotlivé účastníky z komunikace odebírat, volí se signál *user_sig* s hodnotou „leave“. Pokud zvolíme počet klientů 10000, je nastavená počáteční hodnota klientů na sto členů. Pomocí signálu „pridej“ přes tlačítko „SENDTO“ lze uměle zvyšovat počet klientů v nastavených hodnotách 100, 200, 500 a 1000.



Obr. 34: SDL simulátor aplikace

Pomocí tlačítek v levém okně lze simulaci ovládat. Simulaci lze sledovat po krocích, po přechodech nebo nadefinovat bod, ve kterém se simulace zastaví. Do simulace lze také kdykoliv vstupovat a vyslat do procesu libovolný možný signál. Lze ji sledovat v textovém režimu, v režimu SDL (náhled chování systému), či pomocí charakteristik MSC. Ty nás budou zajímat nejvíce.

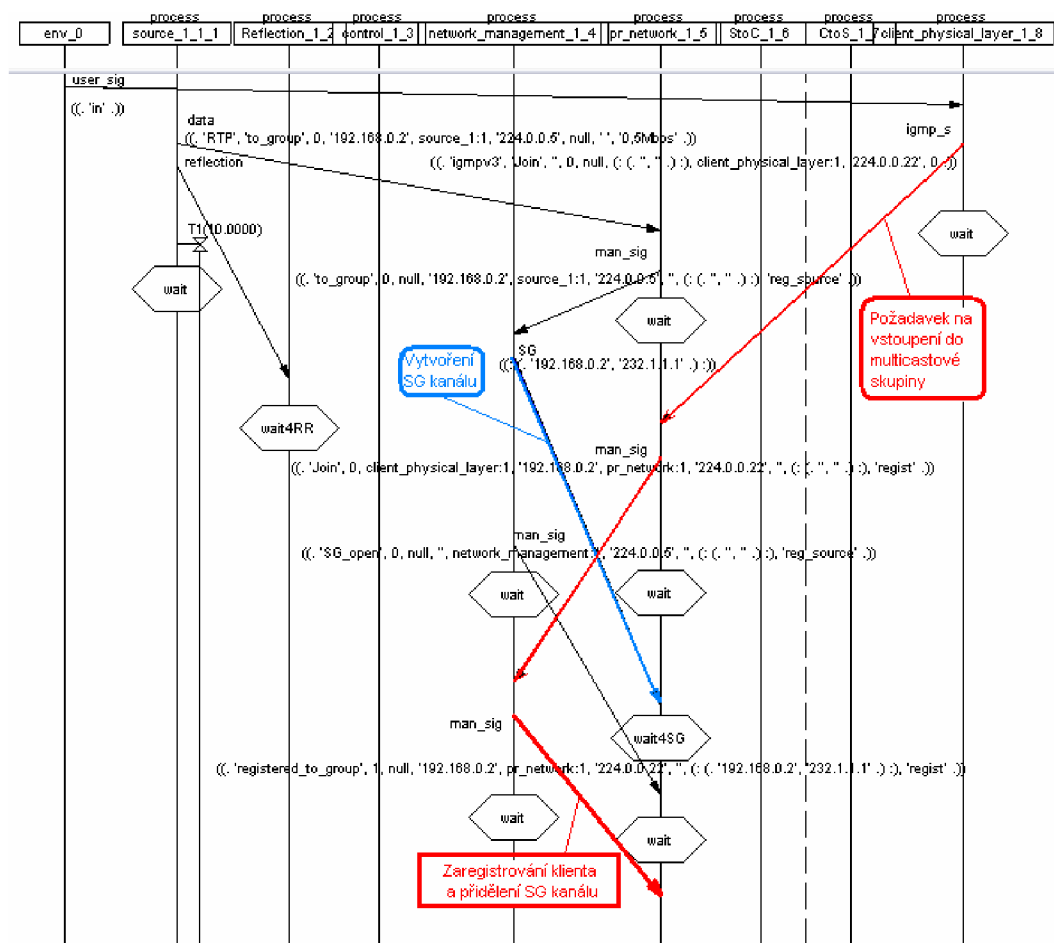
6.2.4 MSC (Message sequence charts)

Jedním z velice efektivních nástrojů programovacího jazyka SDL jsou diagramy MSC, které umožňují přehledně graficky zobrazit časovou posloupnost událostí, které se v daném modelovaném systému vyskytují.

Pomocí těchto diagramů lze sledovat funkce stavových automatů, trasy signálů, i činnost jednotlivých časovačů. Jednotlivými úrovněmi diagramu jsou bloky systémy, které mezi sebou komunikují prostřednictvím signálů.

6.2.4.1 Zahájení vysílání dat a přihlášení účastníka do skupiny

Jak již bylo řečeno, po celkovém nastavení se začíná samotná simulace. První kroky této simulace, tedy hlavní myšlenka první části simulace modelu, je vytvoření samotného SG kanálu (to probíhá v managementu sítě, kde se díky IP adrese zdroje a IP adrese multicastové skupiny daný kanál vytvoří). Na MSC diagramu je to znázorněno modrou barvou. Pro účastníka, který se chce tedy do skupiny připojit a (S,G) kanál je již vytvořen, stačí vyslat zprávu protokolu igmp „join“. Toto je všeobecný požadavek o začlenění do skupiny. Následně router tohoto účastníka registruje a přiděluje mu již vytvořený (S,G) kanál. Na obrázku níže je tato procedura označena modrou barvou. Je-li tedy účastník přítomen v multicastové skupině, může začít samotný přenos RTC/RTCP dat.

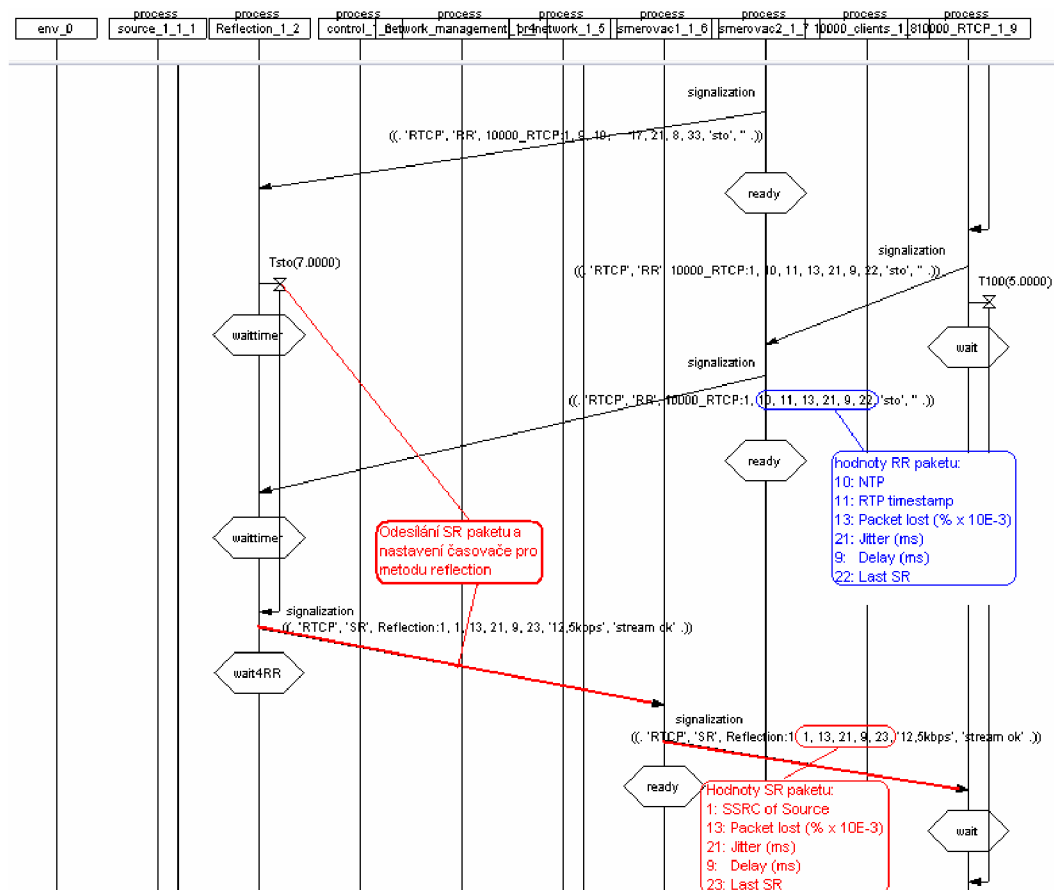


Obr. 35: Přihlášení do skupiny a vytvoření SG kanálu

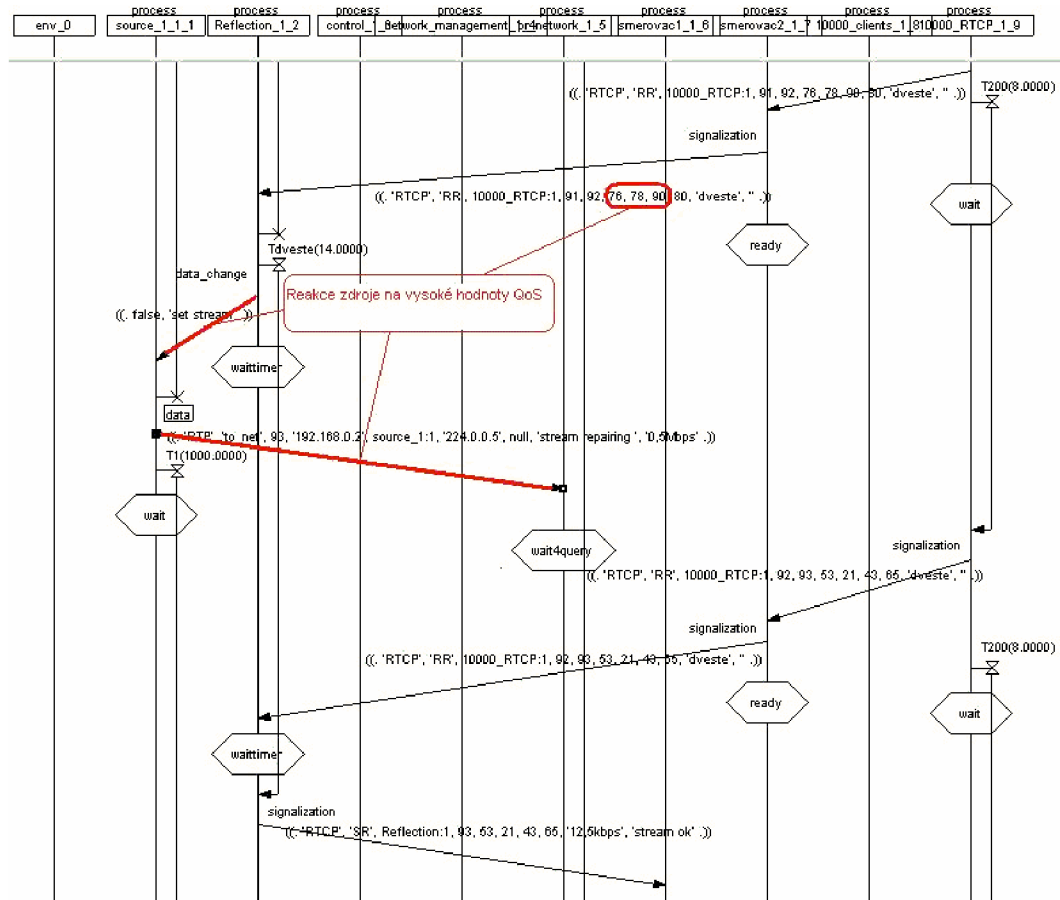
6.2.4.3 Signalizace odesílání sender reportů

Odesílání SR paketů sender reportů závisí na použité metodě odesílání ve zdroji. Metoda reflection je rychlejší pro odpověď, ale častým vysíláním SR paketů by zase mohla zatěžovat síť. Narozdíl od toho metoda summarization vytváří časové zpoždění, ale je šetrnější k šířce pásma, protože ji nezatěžuje častým odesíláním paketů. K samotnému hodnocení obou metod dojdeme později, zde na názorném obrázku si jenom vysvětlíme, jak zobrazená metoda reflection funguje. Zde vidíme, že vypočítaný interval pro odesílání SR je pro skupinu o sto klientech 7s. Po těchto intervalech zdroj odpovídá na přijaté RR. Reálná hodnota je ale v rozmezí kolem 7s. Ta se vypočítává podle reálné velikosti paketu RR. Reálně hodnoty model vypočítává a zaznamenává. V další kapitole se budeme touto analýzou dále zabývat. Na obrázku stojí za povšimnutí i reálné hodnoty RR a SR paketu ohledně nejdůležitějších vlastností vysílání a QoS jako jsou ztrátovost (packet lost), kolísání zpoždění (jitter) a zpoždění (delay). Zdroj tyto hodnoty vyhodnocuje, zaznamenává a analyzuje, a pokud jsou hodnoty pro kvalitu vysílání nedostačující, pokusí se vysílání upravit.

Tuto skutečnost zobrazuje i obrázek 38, kde je znázorněn tento provoz a hodnoty QoS.



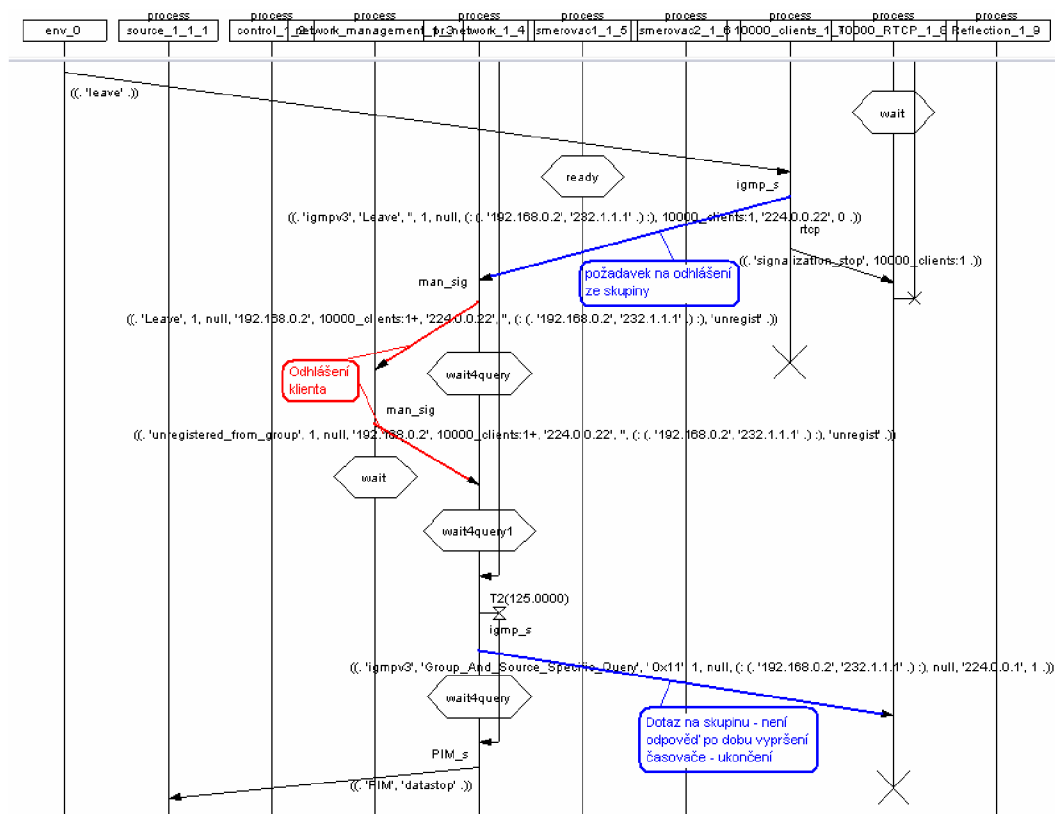
Obr. 37: Odesílání SR paketu s hodnotami QoS



Obr. 38: vyhodnocení RR, analýza hodnot QoS a náprava vysílání

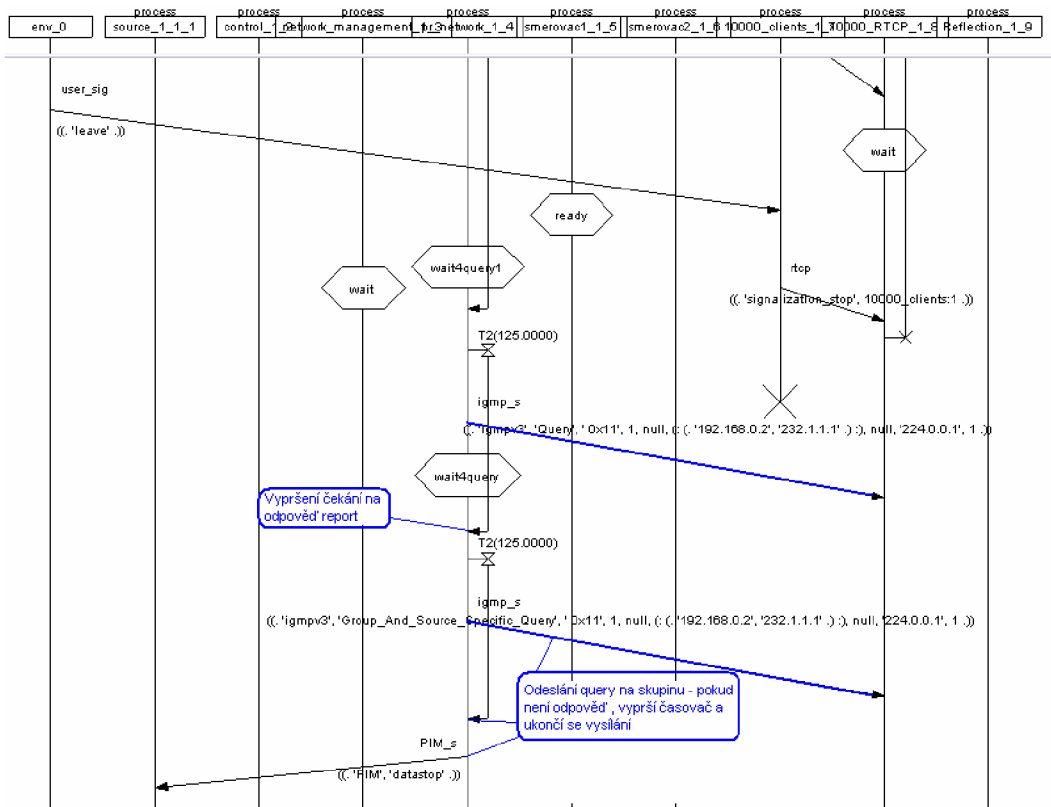
6.2.4.4 Ukončení spojení klientem a odhlášení klienta

Již zmíněnou výhodou využití protokolu IGMPv3 je to, že klient může komunikaci ukončit, kdykoliv chce. To se dá uskutečnit posláním zprávy IGMPv3 typu „Leave_Group“. Jakmile se tedy klient takto rozhodne, odesílá tuto zprávu na router s tím, že už nechce odebírat data a odpojí se. Je tedy na routeru, aby zjistil, zda-li ve skupině není ještě nějaký další klient, který chce data odebírat. To se děje pomocí zprávy IGMPv3 typu „Group_And_Source_Specific_Query“, kde se zjišťuje, zda-li pro daný SSM kanál není ještě nějaký zájemce viz. obrázek dole.



Obr. 39: Leave Group proces

Druhou možností jak lze danou relaci ukončit je ta, že klient přestane odesílat odpovědi na dotazy routeru. To znamená, že o svém požadavku dostávat data neodešle odpověď Report po dobu větší než 125s, což je doba implicitně nastavená pro odesílání dotazů Query. Jakmile router nedostane tuto odpověď do této doby, považuje relaci za ukončenou a odesílá dotaz Qgroup_Specific_Query pro danou skupinu, zda-li tam není nějaký zájemce. Jestliže ano, komunikace probíhá podle pravidel znázorněných výše dále. Tuto druhou možnost zobrazuje následující obrázek.



Obr. 40: Vypršení času pro odpověď na dotaz o odběr dat

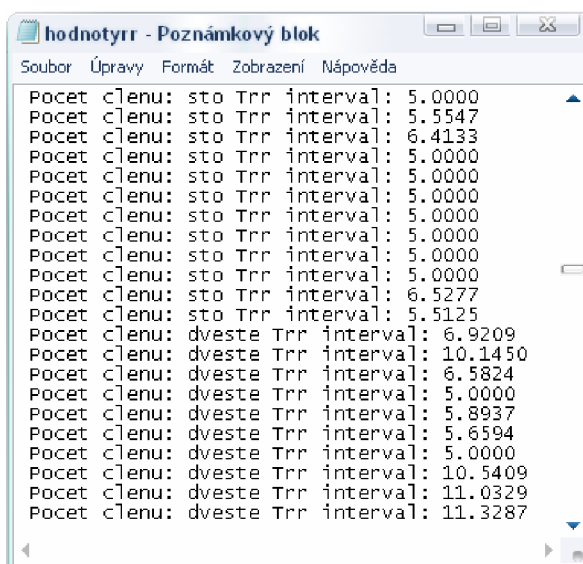
MSC diagramy jsou velice efektivním nástrojem pomáhajícím porozumět funkcím jednotlivých bloků a stavových automatů. Za pomoci nástroje SDL Simulátor lze navíc sledovat chod systému po krocích, což je velice nezbytné pro pochopení činnosti a kontrolu správné funkce navrženého modelu.

7. Praktické využití simulačního modelu

Cíl kapitoly: Pomocí vytvořeného simulačního modelu analyzovat a zpracovat vypočítané hodnoty pro odesílání signalizace za použití metod summarization nebo reflection a porovnat je s teoretickými předpoklady zmíněnými v kapitole 5.

7.1 Simulace odesílání časových intervalů

Výstupem simulačního modelu je textový soubor, obsahující záznam hodnoty intervalu pro odesílání SR a RR reportů pro jednotlivý počet klientů. Výstupní soubor je umístěn přímo v kořenovém adresáři simulačního modelu v programu Telelogic Tau. Struktura výstupního textového souboru je zobrazena na obrázku níže.



Obr. 41: Ukládání reálných hodnot časových intervalů do výstupního souboru

Reálnými hodnotami je zde myšlena skutečná hodnota časového intervalu. Ta byla v minimálním případě zmíněna jako 5s. Skutečností však je, že časový interval se pohybuje v rozmezí násobku (0,5 – 1,5) skutečné hodnoty. Vzhledem k tomu, že minimální hodnota nemůže jít pod zmíněných 5 sekund, můžeme porovnat vypočtené hodnoty v kapitole č.5 se zaznamenanými reálnými hodnotami simulačního modelu, a to pro stejné počty klientů a obě šířky pásma jako v kapitole č. 5, a to pro obě zvolené metody - jak reflection, tak summarization.

7.1.1 Porovnání vypočtených a simulovaných hodnot

Porovnání teoreticky vypočtených hodnot s reálnými hodnotami pro daný počet klientů. Zde je třeba zmínit, že simulace probíhala pro nejvyšší počet klientů 1000, a to proto, že časové intervaly pro tento počet klientů už jsou nesmírně vysoké, a z časové náročnosti proto nebyly simulovány hodnoty časových intervalů pro skupinu klientů o 10000 členech.

Tab. 7: Porovnání reálných a vypočtených hodnot T_{RR} – reflection

Šířka pásma[Mbps]	0,500			
Počet klientů [/]	100	200	500	1000
T_{RR} [s]	5	7,851	19,627	39,253
T_{RRMIN} [s]	5	5	10,159	19,623
T_{RRMAX} [s]	7,352	11,889	29,792	58,329
Šířka pásma[Mbps]	1			
T_{RR} [s]	5	5	9,813	19,627
T_{RRMIN} [s]	5	5	5	10,017
T_{RRMAX} [s]	7,478	7,455	14,989	29,754

Hodnoty časových intervalů T_{RR} metody summarization jsou podobné jako hodnoty metody reflection, proto, zde nebudou uváděny. Můžeme soudit, že hodnoty budou podobné jako v předchozí tabulce.

Dalším výsledkem simulačního modelu bude záznam časových intervalů pro druhou metodu summarization, se změnou podle počtu klientů a šířky pásma.

Tab. 8: Porovnání reálných a vypočtených hodnot T_{RSI} – summarization

Šířka pásma[Mbps]	0,500			
Počet klientů [/]	100	200	500	1000
T_{RSI} [s]	92	207	600	1322
T_{RSIMIN} [s]	46,029	103,529	300,017	661,099
T_{RSIMAX} [s]	137,971	310,489	898,117	1981,765
Šířka pásma[Mbps]	1			
T_{RSI} [s]	46	104	300	662
T_{RSIMIN} [s]	23,114	52,101	150,088	331,241
T_{RSIMAX} [s]	68,899	155,743	449,902	992,615

7.1.2 Porovnání velikosti odeslaných dat RTCP signalizace pro obě metody

Možností simulace bylo také zaznamenávání počtu odeslaných paketů RTCP a jejich velikostí pro jednotlivé velikosti počtu klientů. Velikosti paketů již byly vypočteny v kapitole č.5, proto se zde vychází z těchto hodnot. Velikost objemu odeslaných dat se tedy simuluje v závislosti na čase a počtu odeslaných paketů za tuto dobu. Z této metody můžeme přesněji určit přesný počet odeslaných SR a RR paketů, sečíst jejich velikosti a vypočítat výsledný objem dat přenesený za jednotku času.

Počet posílaných paketů během simulace byl pro jednotlivé metody zaznamenáván a jeho hodnoty jsou uvedeny v tabulce č. 9.

Simulace probíhala po dobu 1 hodiny, tj. 3600s, kdy byl zaznamenáván počet přenesených reportů a v závislosti na jejich velikosti, pak vypočten celkový počet přenesených dat.

Výpočet přenesených dat metody reflection : standardní velikost paketu je 92byte (736b) RR a 60byte (480b) SR, počet přenesených paketů za 1 hodinu 576, šířka pásma 0,5Mbps, počet klientů 100.

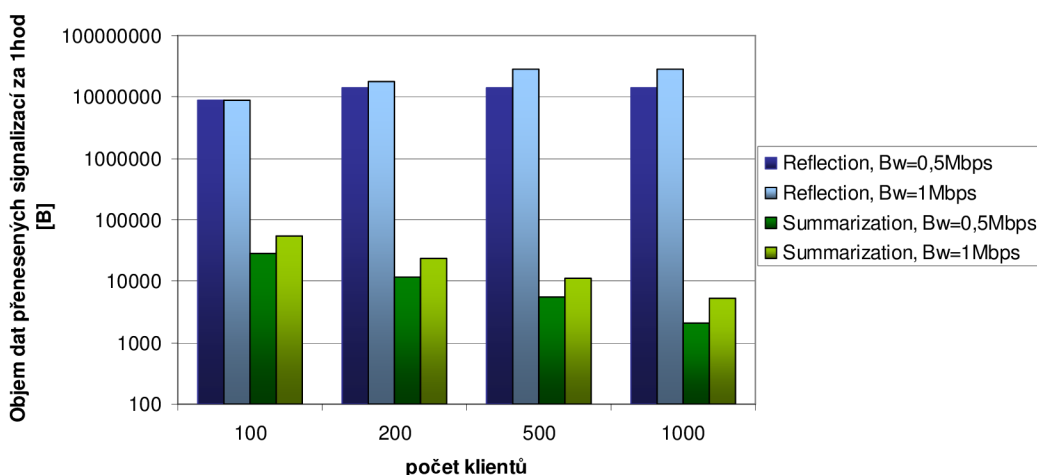
$$\begin{aligned} \text{Objem dat} &= \text{poč. zaznamenaných paketů} \times (PS_{RR} + PS_{SR}) \times n_T \\ \text{Objem dat} &= 576 \times (736 + 480) \times 100 = \underline{70041600\text{bit}} = \underline{8755200\text{byte}} \end{aligned}$$

Tab. 9: Počet odeslaných paketů během simulace a celková velikost přenesených dat

Počet klientů	100	200	500	1000
Šířka pásma [Mbps]	0,5			
Reflection	576	460	186	91
Summarization	39	17	6	2
Objem přenesených dat - Reflection [B]	8755200	13984000	14136000	13832000
Objem přenesených dat - Summarization [B]	27885	11662	5622	2066
Šířka pásma [Mbps]	1			
Reflection	576	576	367	183
Summarization	78	34	12	5
Objem přenesených dat - Reflection [B]	8755200	17510400	27892000	27816000
Objem přenesených dat - Summarization [B]	55770	23324	11244	5165

Tyto vypočtené hodnoty jsou ovšem jenom přibližné, neboť počet přenesených paketů se může lišit, a to díky náhodnému nastavení časových intervalů, který se pohybuje v rozmezí (0,5 – 1,5) násobku jeho hodnoty. Ovšem rozdíly těchto hodnot nejsou vůbec razantní.

Tento výpočet objemu dat během určitého časového intervalu (zde 1 hod.) slouží spíše jako ukázka, jaké je při dané šířce pásma vytížení linek díky signalizaci, která probíhá přenosem paketů sender či receiver reportů. Rozdíly přeneseného objemu dat při simulaci jednotlivých metod signalizace a při volbě šířky pásma 0,5Mbps a 1Mbps zobrazuje obrázek níže.



Obr. 42: Porovnání přeneseného objemu dat za jednotku času

7.1.3 Zhodnocení výsledných simulovaných hodnot

Z výsledků simulace nám tedy vychází že teoretický předpoklad obou metod byl správný, neboť se potvrzuje, že za použití metody reflection nevzniká vysoké zpoždění pro signalizaci, jinými slovy, časové intervaly pro odesílání reportů jsou menší. Bohužel ale díky častému posílání těchto reportů sítí putuje někdy až zbytečně mnoho paketů a pokud bychom například nedisponovali potřebně velkou šířkou pásma, mohlo by docházet k problémům.

Tyto problémy by mohla vyřešit metoda summarization, která díky spojování paketů vytvoří jeden větší spojovaný paket. Díky této metodě a větším časovým prodlevám při odesílání těchto reportů neputuje sítí taková velikost dat jako při metodě reflection, ale mnohem menší. Můžeme tedy tvrdit, že metoda spojování paketů je šetrnější k šířce pásma. Viz Obr.42. Díky velkým časovým prodlevám při odesílání mohou ovšem být informace od klientů v dobu, kdy se dostanou ke zdroji už nepotřebné, a mohou tedy nabývat spíše informativního charakteru.

Tuto nevýhodu dané metody by mohla vyřešit technologie hierarchické agregace klientů do stromové struktury, viz kap. 4.2.3 a 5.4, kde by se předešlo velikému zpoždění, a tudíž by informace od klientů byly při příchodu ke zdroji aktuální a zdroj dokázal na případné problémy včas reagovat.

7.2 Simulace hodnot primárních parametrů QoS

Vzhledem k tomu, že simulační model generuje pro odesílané RR pakety i hodnoty nejdůležitějších veličin QoS, jako jsou ztrátovost (Packet lost), zpoždění (Delay) a kolísání zpoždění (Jitter), je možné zaznamenávat i tyto hodnoty a třeba je později i analyzovat. V této aplikaci ale slouží jen pro rozhodování zdroje, jak přizpůsobit vysílání multimediálních dat. Uvedenou skutečnost zobrazují obrázky č.37 a č.38.

Zde se tedy nabízí mnoho dalších možností, jak lze dále rozšiřovat simulační model a komunikaci mezi klienty a zdrojem. To ovšem není zadáním této práce, proto zde tato možnost nebyla dále prohlubována.

8. Závěr

V dnešní době moderního internetu, kdy se multimediální komunikace rozvíjí obrovským tempem stále vpřed máme širokou škálu možností, jak přes tuto síť přenášet, sdílet či čerpat multimediální data. A to prostřednictvím videokonferencí, rádiového vysílání prostřednictvím internetu nebo IPTV, kdy jsou data šířena vždy od určitého zdroje dat nebo více zdrojů přes vnější prostředí k určitému spotřebiteli či skupině klientů. Proto, abychom dosáhli úspěšného přenosu dat od zdroje ke klientům, který není chybově degradovaný, časově zpožděný či nijak jinak znehodnocený, je třeba dodržovat určitá pravidla přenosu a řídit se jimi tak, aby byl klient- uživatel se službou spokojen. Z těchto důvodů je třeba zajímat se o tzv. signalizaci mezi zdrojem dat a jejich spotřebitelem, tzn. výměnou řídicích zpráv mezi zdrojem a klientem tak, aby se případné neshody mezi těmito skupinami daly vyřešit.

V první části práce - úvodu - je rozebrána teorie týkající se technologie multicast. Jednotlivé dílčí body této technologie, hlavní myšlenky a nejdůležitější části jsou z důvodů přehlednosti členěny do kapitol č.1 až č.4. V první kapitole je rozebrána technologie multicast. Ve druhé kapitole je rozebrána blíže její specifikace Source-Specific Multicast SSM, ve třetí následuje rozbor jednotlivých používaných protokolů, které tato metoda využívá. Ve čtvrté kapitole, která se zabývá teorií komunikace mezi zdrojem dat a klienty, je uvedena problematika vysílání a signalizace při výměně multimediálních dat protokoly RTP a RTCP, jejich problematika v použití konferenčního spojení, nebo multimediální relace a návrh dalšího řešení či postupu při problémech spojenými s těmito mechanismy. Poslední kapitola první části, kapitola č.5, uvádí návrh teoretického modelu komunikace zdroj- příjemci. Pomocí vzorců v předchozích kapitolách jsou vypočítány stěžejní hodnoty, ze kterých bude simulační model v praktické části vycházet., Tento model je následně v druhé části, praktické části diplomové práce, navržen v programovém prostředí jazyka SDL tak, aby se pokusil upřesnit získané informace a podložil je reálnou skutečností zdokumentovanou a vyobrazenou v MSC diagramech a v hodnotách získaných během použití simulačního modelu v praktickém využití.

V praktické části práce je nejprve zahrnut úvod do programového prostředí SDL, jeho základní stavba a teoretické poznatky tohoto specifického programovacího jazyka. Dále už se práce zabývá konkrétním přístupem k samotné tvorbě aplikace, jejím rozбором, analýzou a vysvětlením jednotlivých částí, a vůbec celou skladbou funkční aplikace s doloženými ukázkami. Výstupem aplikace jsou MSC diagramy zobrazující simulaci komunikace mezi jednotlivými aplikačními částmi. Tyto MSC diagramy jsou rozděleny z důvodů přehlednosti do několika částí. (viz. obr.35 až 40). Celá komunikace je takto podle obrázků rozdělena do přehledných etap komunikace, ze kterých je zřetelnější pochopit rozebíranou problematiku.

Dalším krokem praktické části byla simulace aplikace a její chování v závislostech na počtu komunikujících klientů, či použité metodě signalizace. Výsledky simulace byly zaznamenány a následně zpracovány v poslední kapitole praktické části., kapitole č.7, která vysvětluje, vyhodnocuje zpracovaná data a ze získaných výsledků je zpracován výsledek a charakter simulace. Významným článkem ve zhodnocení praktické části jsou tabulky 7 až 9 a obr. 42, které zobrazují chování probírané komunikace v reálné situaci a ukazují charakter použitých služeb a metod. Dosažené výsledky simulačního modelu SSM, vytvořeného v praktické části, korespondují se znalostmi získaných v teoretické části práce a splňují vytyčené cíle.

Použitá literatura:

- [1] DOLDI L. *SDL Illustrated, Visually design executable models*, Paragraphic 2001, 269 stran, ISBN 2-9516600-0-6
- [2] Ing. HERMAN, I., CSc., *TECHNKAPARALERNÍCH PROCESŮ*, návod do cvičení, VUT Brno, FEKT, Ústav telekomunikací, 54 stran.
- [3] Ing. KOMOSNÝ, D., Ph.D., Ing. MURSHED, G., A., A , *Jazyk SDL* [online].c2004, poslední revize 30.3.2004 [cit.2007-11-05], dostupné z: <<http://www.elektrorevue.cz/clanky/04018/>>
- [4] BACCALA, B., *RTP/RTCP control protocol* [online].c1997, poslední revize 17.1.2002 [cit.2008-2-3], dostupné z: <<http://www.networksorcery.com/enp/protocol>>
- [5] Ing. KOMOSNÝ, D., Ph.D., *Nové směry vývoje protokolu RTP/RTCP pro rozsáhlé konference v Internetu* [online].c2004, poslední revize 27.10.2004 [cit.2007-03-05], dostupné z: <<http://www.elektrorevue.cz/clanky/04052/index.html#52>>
- [6] BHATTACHARYA, S., *An Overview of Source-Specific Multicast (SSM)* [online].c2003, poslední revize 27.6.2003 [cit.2008-22-02], dostupné z: <<http://www.ietf.org/rfc/rfc3569.txt>>
- [7] FILIP, O., *Úvod do IP multicastu* [online].c2004, poslední revize 10.9.2004 [cit.2007-10-29], dostupné z: <<http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-prvni/>>
- [8] FILIP, O., *Úvod do IP multicastu (díl druhý)* [online].c2004, poslední revize 20.9.2004 [cit.2007-10-29], dostupné z: <<http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-druhy/>>
- [9] FILIP, O., *Úvod do IP multicastu (díl třetí)* [online].c2004, poslední revize 8.10.2004 [cit.2007-10-29], dostupné z: <<http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-treti/>>
- [10] FILIP, O., *Úvod do IP multicastu (díl čtvrtý)* [online].c2004, poslední revize 29.10.2004 [cit.2007-10-29], dostupné z: <<http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-ctvrty/>>
- [11] FILIP, O., *Úvod do IP multicastu (díl pátý)* [online].c2004, poslední revize 23.11.2004 [cit.2007-10-29], dostupné z: <<http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-paty/>>
- [12] OTT, J., *RTP and RTCP overview* [online].c1998, poslední revize 15.1.1998 [cit.2007-12-29], dostupné z http://www.cis.ohio-state.edu/~jain/cis788-97/ip_multimedia/index.htm

-
- [13] Ing. KOMOSNÝ, D , Doc. Ing. NOVOTNÝ, V., *Tree Structure for Source-Specific Multicast Feedback Aggregation*, [online].c2007, poslední revize 29.10.2004 [cit.2008-13-03], dostupné z:
<<http://adela.utko.feec.vutbr.cz/projects/publications/#1>>
- [14] Ing. KOMOSNÝ, D , Doc. Ing. NOVOTNÝ, V., *Optimization of Large-Scale RTCP Feedback Reporting in Fixed and Mobile Network* [online].c2007, poslední revize 29.10.2004 [cit.2008-14-03], dostupné z:
<<http://adela.utko.feec.vutbr.cz/projects/publications/#3>>
- [15] JACOBSON, V., *RTP: A Transport Protocol For Real-Time Applications* [online].c1996, [cit.2008-02-10], dostupné z:
<<http://www.freesoft.org/CIE/RFC/1889/index.htm> >