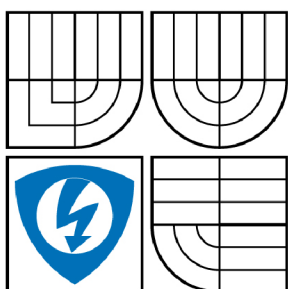




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## PROTOKOL TTP PRO SPRÁVU HIERARCHICKÝCH STROMŮ ZPĚTNÉ VAZBY RTCP KANÁLU

TTP PROTOCOL FOR MANAGING HIERARCHY TREES OF RTCP FEEDBACK CHANNEL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAKUB MÜLLER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RADIM BURGET

BRNO 2008

Vložit zadání

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Bc. Jakub Müller  
Bytem: Kunčice 239, 56151, Letohrad - Kunčice  
Narozen/a (datum a místo): 7.7.1984, Ústí nad Orlicí

(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: Protokol TTP pro správu hierarchických stromů zpětné vazby  
RTCP kanálu

Vedoucí/školitel VŠKP: Ing. Radim Burget

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 zpracování 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

Protokol TTP pro správu hierarchického stromu zpětné vazby RTCP kanálu představuje mechanismus, který realizuje přenos velkého množství informací od uživatelů „úzkým“ zpětným kanálem. Velkým množstvím není míněno tisíce uživatelů, ale spíše až miliony, kterých mohou služby jako je IPTV dosáhnout. K tomuto účelu využívá metody sumarizace informace od části uživatelů v definovaných uzlech sítě a následný přenos zpráv a další případnou sumarizaci v nadřazených uzlech sestaveného hierarchického stromu zpětného kanálu. Obě metody umožňují především redukci množství dat, což se následně promítne do výrazného většího množství informace, které je zpětným kanálem přeneseno. Pro správnou funkčnost je třeba vždy zjistit aktuální pozici uživatele a podle ní vypočítat nejbližší sumarizační uzel, ke kterému bude svá data odesílat ke zpracování. Pro realizaci zpětného kanálu existuje hned několik možností, z kterých je představen především asynchronní způsob přenosu zpráv, sestavení a následná správa tohoto kanálu.

## Klíčová slova

Zpětná vazba; agregace; sumarizace; IPTV; TTP; RSI;

## **Abstract**

TTP protocol for managing hierarchy trees of the RTCP feedback channel represents the mechanism for transferring the big amount of data from end users via the “narrow” feedback channel. We are not speaking about thousands of users but about millions of users, which are using services like IPTV. The method of a data summarization is used for this purpose in selected network nodes. The summarized message is transferred and summarized again in higher levels of the hierarchical tree. Both methods allow reducing the amount of data and help to increase information content transferred via the feedback channel. Finding of the correct end user position in a network is also very important aspect. The user must be able to find the closest summarization node with this information and starts sending out the messages to this node for processing. There are several methods for constructing and managing the asynchronous feedback channel that are introduced in this work.

## **Keywords**

Feedback; aggregation; summarization; IPTV; TTP; RSI;

## Bibliografická citace

MÜLLER, J. *Protokol TTP pro správu hierarchických stromů zpětné vazby RTCP kanálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 75 s. Vedoucí diplomové práce Ing. Radim Burget.

## Prohlášení

Prohlašuji, že svoji diplomovou práci na téma protokol TTP pro správu hierarchických stromu zpětné vazby RTCP kanálu jsem vypracoval samostatně pod vedením vedoucího diplomové práce, 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 práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

(podpis autora)



## Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Radimu Burgetovi za odborné vedení a cenné rady při zpracování diplomové práce a dále pak Ing. Danu Komosnému, PhD. za poskytnutí možnosti práce na projektu. V neposlední řadě také děkuji rodině a všem, kteří mě v průběhu celého studia podporovali.

V Brně dne .....

.....

(podpis autora)

## Seznam zkratek

ASM	Any-Source Multicast
DoS	Denial of Service
FT	Feedback Target
FTD	Feedback Target Definition packet
FTI	Feedback Target Information packet
FTM	Feedback Target Manager
FTS	Feedback Target Specification packet
GNP	Global Network Positioning
GPS	Global Positioning System
HA	Hierarchic aggregation
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IPTV	Internet Protocol Television
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
LM	LandMark
QoS	Quality of Service
R	Receiver
RFC	Request for Comments
RFT	Root Feedback Target
RR	Receiver Report
RSI	Receiver Summary Information packet
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTT	Round-Trip Time
S	Sender
SR	Sender Report
SSM	Source Specific Multicast
TTP	Tree Transmission Protocol
UML	Unified Modeling Language

## Obsah:

ÚVOD .....	15
1 PROTOKOL RTP/RTCP .....	17
2 PROTOKOL PRO VYTVÁŘENÍ ZPĚTNÉHO KANÁLU .....	19
2.1 Způsoby zvýšení počtu příjemců .....	21
2.1.1 Sumarizace .....	21
2.1.2 Sumarizační paket (RSI) .....	22
2.1.3 Stromová struktura (Hierarchická agregace) .....	24
2.2 Zajištění lokalizace příjemce .....	26
2.3 Příslušnost k sumarizačnímu bodu .....	29
2.4 Členové zpětného kanálu .....	30
2.4.1 Vysílač (S) a příjemce (R) .....	30
2.4.2 Sumarizační uzel (FT) a Kořenový uzel (RFT) zpětného kanálu .....	31
2.4.3 Manažer uzlů zpětné vazby (FTM) .....	31
2.5 Pakety pro sestavení a správu zpětného kanálu .....	33
2.5.1 Specifikační paket zpětné cesty (FTS) .....	35
2.5.2 Definiční paket (FTD) .....	38
2.5.3 Informační paket (FTI) .....	39
3 SESTAVENÍ HIERARCHICKÉHO ZPĚTNÉHO KANÁLU PRO IPTV .....	40
3.1 Matematický aparát .....	40
3.2 Inicializace stromu .....	44
3.3 Možnosti realizací .....	45
3.3.1 Asynchronní zpětný kanál .....	47
3.3.2 Synchronní zpětný kanál .....	49
3.3.3 Využití více vrstev hierarchického stromu .....	50
3.3.4 Definování velikosti sumarizačního paketu .....	52
4 PRAKTICKÁ REALIZACE ZPĚTNÉHO KANÁLU .....	54
4.1 Simulační knihovna JSimLib3 .....	55
4.1.1 Základní struktura knihovny .....	55
4.1.2 Komunikace mezi procesy .....	56
4.2 Sekvenční diagramy .....	57
4.2.1 Sestavení zpětného kanálu .....	57
4.2.2 Využití zpětného kanálu uživateli .....	59

4.3	Stavové diagramy .....	61
4.3.1	Zpracování paketu .....	61
4.3.2	Stavový diagram FT .....	62
4.3.3	Stavový diagram FTM .....	63
4.3.4	Stavový diagram Přijímače.....	64
4.3.5	Stavový diagram stavby hierarchického stromu .....	64
4.4	Řešení potíží při vývoji.....	66
4.5	Volba mechanismu pro tvorbu zpětného kanálu.....	67
4.5.1	Požadavky zákazníka .....	67
4.5.2	Požadavky a možnosti poskytovatele IPTV služby.....	68
4.5.3	Asynchronní kanál s/bez parametrů RTCP.....	68
4.5.4	Synchronní kanál.....	69
	ZÁVĚR.....	70

## Obrázky

OBR. 1.1 - ZÁVISLOST UŽIVATELŮ NA ŠÍŘCE PÁSMU.....	18
OBR. 2.1 - HIERARCHICKÁ STRUKTURA ZPĚTNÉHO KANÁLU.....	20
OBR. 2.2 - STRUKTURA INFORMAČNÍHO PAKETU RTCP POSÍLANÉHO K VYSÍLAČI.....	21
OBR. 2.3 - ZÁKLADNÍ STRUKTURA SUMARIZAČNÍHO PAKETU (RSI).....	23
OBR. 2.4 - PŘÍKLAD STRUKTURY VNOŘENÝCH ČÁSTÍ RSI.....	23
OBR. 2.5 - DATOVÝ TOK S HIERARCHICKOU AGREGACÍ (A) A BEZ NÍ (B).....	24
OBR. 2.6 - DOBA ODEZVY KE VZDÁLENÉMU UZLU V SÍTI.....	26
OBR. 2.7 - POMĚR ODEZVY.....	27
OBR. 2.8 – VEKTORY POZIČNÍCH BODŮ.....	28
OBR. 2.9 - PROCES VYTVÁŘENÍ "KOŠŮ".....	29
OBR. 2.10 - PRÁCE (FTM) PŘI VÝPADKU SUMARIZAČNÍHO UZLU.....	32
OBR. 2.11 - HLAVIČKA VŠECH PAKETŮ TTP.....	33
OBR. 2.12 - UML DIAGRAM TŘÍD.....	34
OBR. 2.13 - FEEDBACK TARGET SPECIFICATION PACKET (FTS).....	35
OBR. 2.14 - SUBBLOK FT.....	36
OBR. 2.15 - SUBBLOK LM.....	37
OBR. 2.16 – DEFINIČNÍ PAKET UZLU (FTD).....	38
OBR. 2.17 – INFORMAČNÍ PAKET (FTI).....	39
OBR. 3.1 - ÚVODNÍ INICIALIZACE VRSTEV HIERARCHICKÉHO STROMU.....	44
OBR. 3.2 – OPTIMALIZACE ZPĚTNÉHO KANÁLU.....	46
OBR. 3.3 - POČET (FT) PRO SYNCHRONNÍ A ASYNCHRONNÍ ZPĚTNÝ KANÁL.....	48
OBR. 3.4 - POSUNUTÍ INTERVALU GENEROVÁNÍ A ODESLÁNÍ RSI ZPRÁV U VYŠŠÍCH VRSTEV HIERARCHICKÉHO STROMU.....	49
OBR. 3.5 - ROZPTYL POČTU PŘÍCHOZÍCH RSI ZPRÁV.....	50
OBR. 3.6 - ROZDÍL POČTU SUMARIZAČNÍCH UZLŮ (FT).....	51
OBR. 3.7 - VELIKOST RSI PAKETU.....	53
OBR. 4.1 - STRUKTURA KNIHOVNY V UML NOTACI.....	55
OBR. 4.2 - KOMUNIKUJÍCÍ PROCESY V RÁMCI SUMARIZAČNÍHO UZLU (FT).....	56
OBR. 4.3 - SESTAVENÍ HIERARCHICKÉHO STROMU ZPĚTNÉHO KANÁLU.....	58
OBR. 4.4 - KOMUNIKACE MEZI PRVKY ZPĚTNÉHO KANÁLU PŘI VYTVÁŘENÍ HIERARCHICKÉHO STROMU.....	58
OBR. 4.5 - PŘENOS RR PAKETŮ OD KONCOVÝCH UŽIVATELŮ.....	59
OBR. 4.6 - PŘENOS ZPRÁV PŘES HIERARCHICKÝ STROM.....	60
OBR. 4.7 – ZPRACOVÁNÍ PAKETŮ FTI,FTD A FTS PO JEJICH PŘÍJMU.....	61
OBR. 4.8 - STAVOVÝ DIAGRAM FT.....	62
OBR. 4.9 - STAVOVÝ DIAGRAM FTM.....	63
OBR. 4.10 - STAVOVÝ DIAGRAM PŘIJÍMAČE.....	64
OBR. 4.11 - DIAGRAM ALGORITMU SESTAVENÍ H-VRSTVÉHO STROMU.....	65
OBR. 4.12 – DRUHY MULTICASTOVÉHO VYSÍLÁNÍ.....	66

## Tabulky

TAB. 1 - SEZNAM VEKTORŮ.....	28
------------------------------	----

## Rovnice

ROVNICE 1.1 - ŠÍŘKA PÁSMY ZPĚTNÉHO KANÁLU .....	17
ROVNICE 1.2 - ŠÍŘKA PÁSMY DOPŘEDNÉHO KANÁLU .....	17
ROVNICE 1.3 - INTERVAL ODESÍLÁNÍ RR ZPRÁV .....	17
ROVNICE 1.4 - INTERVAL ODESÍLÁNÍ SR ZPRÁV .....	18
ROVNICE 2.1 - OBECNÝ VZOREC PRO VÝPOČET VELIKOSTI SUMARIZAČNÍHO PAKETU .....	22
ROVNICE 2.2 - VELIKOST ŠÍŘKY PÁSMY ZPĚTNÉHO KANÁLU PŘI POUŽITÍ RTCP PROTOKOLU .....	25
ROVNICE 2.3 - VELIKOST ŠÍŘKY PÁSMY ZPĚTNÉHO KANÁLU PŘI VYUŽITÍ HIERARCHICKÉ AGREGACE .....	25
ROVNICE 3.1 - CELKOVÉ ZPOŽDĚNÍ PRŮCHODU ZPRÁVY HIERARCHICKÝM STROMEM ZPĚTNÉ VAZBY .....	41
ROVNICE 3.2 - POČET FT V PŘÍSTUPOVÉ VRSTVĚ HIERARCHICKÉHO STROMU .....	41
ROVNICE 3.3 - POČET FT VE 2 VRSTVĚ HIERARCHICKÉHO STROMU .....	42
ROVNICE 3.4 - POČET FT V LIBOVOLNÉ H-VRSTVĚ HIERARCHICKÉHO STROMU .....	42
ROVNICE 3.5 - POČET FT VE 3VRSTVĚ HIERARCHICKÉHO STROMU .....	42
ROVNICE 3.6 - OPTIMALIZACE ČASU PRO 3VRSTVÝ HIERARCHICKÝ STROM .....	42
ROVNICE 3.7 - OPTIMALIZOVANÁ INTERVAL ODESÍLÁNÍ RSI ZPRÁV MEZI 1-2 A 2-3 VRSTVOU HIERARCHICKÉHO STROMU .....	43
ROVNICE 3.8 - OPTIMALIZOVANÝ INTERVAL ODESÍLÁNÍ ZPRÁV SHODNÝ PRO VŠECHNY 3VRSTVY HIERARCHICKÉHO STROMU .....	43
ROVNICE 3.9 - OPTIMALIZOVANÝ INTERVAL POSÍLÁNÍ RSI ZPRÁV MEZI VRSTVAMI LIBOVOLNÉHO HIERARCHICKÉHO STROMU SE ZACHOVÁNÍM PŘÍSTUPOVÉHO INTERVALU .....	43
ROVNICE 3.10 – SPOLEČNÝ OPTIMALIZOVANÝ INTERVAL POSÍLÁNÍ VŠECH ZPRÁV MEZI VRSTVAMI LIBOVOLNÉHO HIERARCHICKÉHO STROMU .....	43
ROVNICE 3.11 - PODROBNÝ VZOREC PRO VÝPOČET VELIKOSTI RSI ZPRÁVY .....	52

## Přílohy

PŘÍLOHA 1 - SEKVENČNÍ DIAGRAM PROVOZU VE ZPĚTNÉM KANÁLE .....	73
PŘÍLOHA 2 - KOMPLETNÍ DIAGRAM TŘÍD TTP PAKETŮ .....	74

## ÚVOD

V posledních letech se tvář internetu značně změnila a neustále se mění. Na konci 20. století fungoval internet převážně jako rychlé informační medium, pomocí kterého se uživatelé mohli dostat k nejnovějším zprávám, posílat si emailové zprávy a v neposlední řadě sdílet menší soubory pomocí protokolu FTP. Pro tyto úkony bylo velice typické vytáčené připojení pomocí klasického analogového modemu. Postupem času však nároky uživatelů rostly a ti požadovali více interaktivní zábavy. Pro tyto požadavky však bylo potřebné vlastnit kvalitní a hlavně rychlé připojení. S rozvojem a dostupností nových technologií pro přístup koncových účastníků k internetu, jako jsou ADSL nebo bezdrátové připojení pomocí WiFi, jim byla tato možnost dána za přijatelnou cenu. Rychlost připojení stoupla v průměru až 40ti násobně a uživatelé tak vyměnili „nudné“ textové zprávy za internetovou telefonii (Voice over IP, VoIP), statické internetové stránky za stránky s podporou dynamických animací a v neposlední řadě začali přenášet soubory o velikosti až stovek MB dat. Právě tento typ komunikace má v poslední době dominantní úlohu a jeho kouzlo objevuje stále více uživatelů. Úloha serveru jako úložiště a databáze dat pro uživatele mizí a přesouvá se právě mezi účastníky.

S dostupností kvalitnějšího a hlavně rychlejšího připojení přichází i další možnost využití služeb, které jsou tak typické pro náš každodenní život. Jedná se především o poslech rádia a příjem televizního signálu. Internetová rádia fungují již delší dobu a je asi málo těch, co někdy s touto službou nepřišli do kontaktu nebo jí dokonce pravidelně nevyužívají. Existuje již také celá řada televizních stanic, které nabízejí své služby pomocí sítě internetu. Jedná se o službu, která nese zkratku IPTV (Internetová televize). Obě tyto služby jsou zajištěny buď formou unicastu nebo moderněji multicastu, kde existuje jeden zdroj a několik stovek až tisíc příjemců. Data jsou vysílána jen jednou do tzv. multicastové skupiny. Jelikož unicast je, hlavně v případě internetové televize (IPTV), značně datově náročný, je v případě IPTV prosazován především multicast, kdy existuje pouze jedno spojení od zdroje a replikace paketů ke všem příjemcům je přenechána na směrovačích. Všechny multimediální služby používají pro distribuci dat převážně protokol RTP/RTCP [1], který byl právě pro tento způsob komunikace navržen. Přenáší data, která jsou citlivá buď na zpoždění (případ VoIP) nebo na interval potvrzování o kvalitě příjmu vysílaných dat (případ IPTV). Cílem mé práce není změna RTP protokolu, který ostatně i v dnešních podmínkách funguje velice dobře, ale především podpora RTCP protokolu, který při účasti velkého množství uživatelů naráží na strop své funkčnosti. Velké množství uživatelů totiž generuje velké množství dat, které je třeba přenést, a tím se zvyšuje doba odezvy od jednoho uživatele. Pokud by byla zvolena zcela nová koncepce přenosu, znamenalo by to vytvoření nových aplikací, jelikož ty stávající by nemohly být využívány. Účelem této práce je poskytnout kvalitu služeb (QoS) pro relace s velkým počtem uživatelů v řádech stovek tisíců uživatelů.

Tato práce je obsahově rozdělena do několika částí. Kapitola 1 je zaměřena na představení problému s relací RTP/RTCP. U velkého počtu uživatel totiž dochází k požadavku na příliš velkou šířku pásma, kterou daná topologie může poskytnout. V kapitole 2 je pak představen nově vytvořený protokol, který má za úkol řešit potíže popsané v kapitole první. Jsou zde představeni noví členové a druhy zpráv, kteří zajišťují chod celého protokolu, společně s vysvětlením jejich funkcí. Kapitola 3 obsahuje rozvahu nad problematikou vytvoření optimálního zpětného kanálu realizovaného pomocí hierarchického stromu. Dále jsou zde představeny matematické vzorce potřebné k následné realizaci. Kapitola 4 je pak věnována samotnému návrhu funkčnosti a praktické realizaci poznatků definovaných v kapitolách 1, 2 a 3.



# 1 PROTOKOL RTP/RTCP

RTP/RTCP je vlastně dvojice protokolů, která byla vyvinuta za účelem doručování multimediálních dat citlivých na kvalitu služby. Kvalitou služby se rozumí pro každou službu něco jiného. U telefonování se jedná především o zpoždění, u přenosu dat zase o chybovost kanálu nebo u rádia či televize, kde je naopak důležitým kritériem plynulost příjmu multimediálních dat. RTP protokol se stará o přenos a doručení dat od vysílače k příjemci a protokol RTCP pak zajišťuje informování o kvalitě přenášených dat v obou směrech (od zdroje k příjemci a i naopak). Z tohoto důvodu jsou zprávy protokolu RTCP členěny do dvou kategorií na zprávy Sender Report (SR) a Receiver Report (RR). Zprávy SR jsou posílány od vysílače (Sender, S) k příjemci (Receiver, R) a zprávy RR jsou posílány od (R) k (S). Podrobnou specifikaci naleznete v literatuře [1]. K přenosu dat využívá protokol RTP/RTCP většinou služeb transportního protokolu UDP a proto musí kvalitu služeb (QoS) do značné míry zabezpečit vlastními silami. Šířka pásma celé relace je rozdělena v poměru 95:5 (RTP:RTCP), kde se dále kanál pro RTCP dělí v poměru 3:1 (RR:SR). Z tohoto vyplývá, že 95% celého kanálu určeného pro danou relaci zabírá samotný přenos multimediálních dat pomocí RTP protokolu, 3,75% spotřebuje přenos RR zpráv od přijímačů (R) ke zdroji (S) a 1,25% připadá na SR kanál od zdroje (S) k přijímačům (R). Pro přehlednost je přiložena matematická interpretace těchto vztahů, Rovnice 1.1 a Rovnice 1.2. Právě v tomto rozdělení spočívá největší úskalí pro aplikace, které předpokládají obrovské množství uživatelů v řádu stovek tisíců. RR pakety se totiž generují pro každého účastníka samostatně a jsou odesílány přímo k (S). Aby bylo umožněno odeslání informace od všech účastníků, každý (R) si generuje interval odesílání RR zprávy podle aktuálního počtu členů v relaci (Rovnice 1.3 a Rovnice 1.4). Pokud tedy roste počet uživatelů, roste stejnou rychlostí i zpoždění pro odeslání RR zpráv (Obr. 1.1). Dokonalý příklad poskytuje internetová televize (IPTV), která je hlavní motivací pro vznik této práce.

$$BW_{RR} = 5\% \cdot 75\% \cdot BW \quad \left[ \frac{\text{bit}}{\text{s}} \right] \quad (1.1)$$

$$BW_{SR} = 5\% \cdot 25\% \cdot BW \quad \left[ \frac{\text{bit}}{\text{s}} \right] \quad (1.2)$$

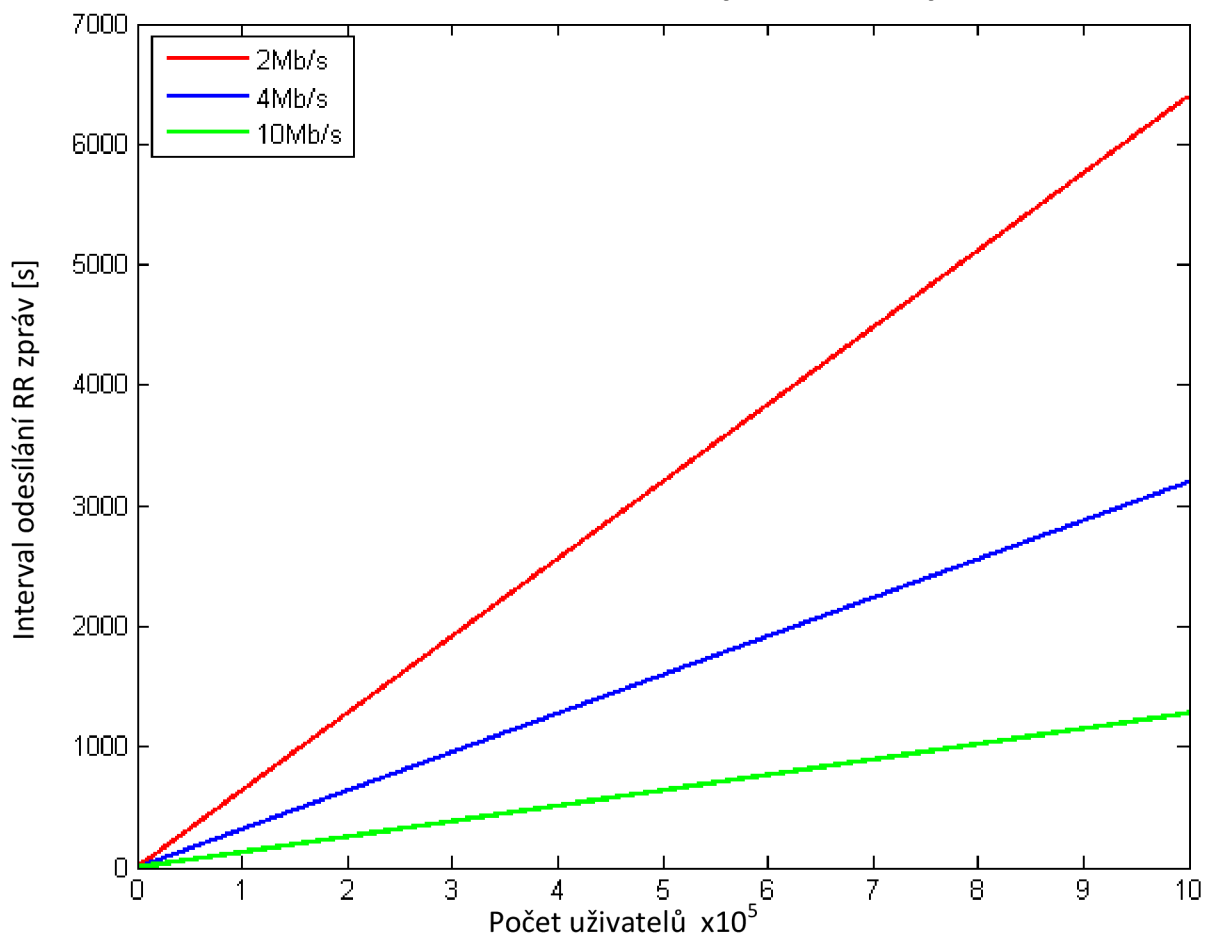
$$T_{RR} = \frac{PL_{RR} \cdot N}{BW_{RR}} \quad [\text{s}] \quad (1.3)$$

$$T_{SR} = \frac{PL_{SR} \cdot M}{BW_{SR}} \quad [s] \quad (1.4)$$

kde

$BW_{RR/SR}$	[bit/s]	značí šířku páska /ve zpětném kanále	/v multicastovém kanále
$M$	[-]	definuje počet vysílačů	
$N$	[-]	definuje počet uživatelů	
$T_{RR/SR}$	[s]	interval odeslání RR/SR zprávy	
$PL_{RR/SR}$	[bit]	je velikost paketu RR/SR	

**Závislost intervalu odesílání RR zpráv na šířce pásma**



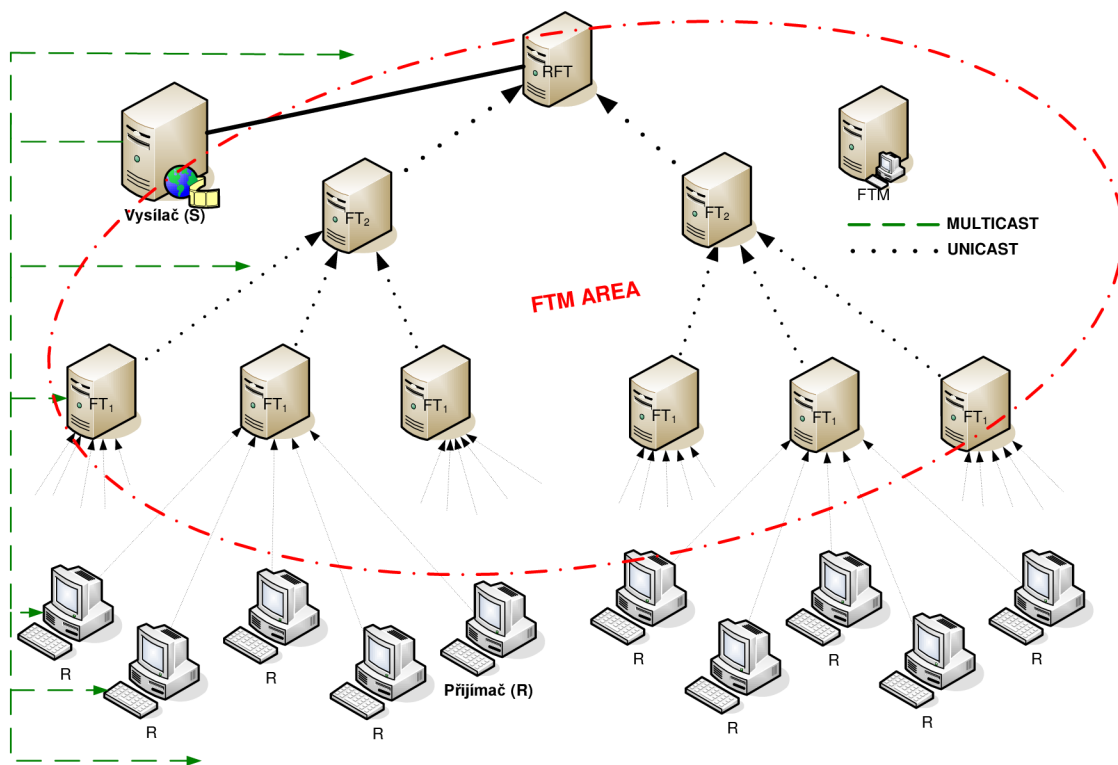
**Obr. 1.1 - Závislost uživatelů na šířce pásma**

## 2 PROTOKOL PRO VYTVÁŘENÍ ZPĚTNÉHO KANÁLU

Protokol TTP je zkratkou plného názvu Tree Transmission Protocol [5]. Jeho autory jsou Vít Novotný a Dan Komosný (oba FEKT VUT Brno). Jedná se samostatný protokol navržený pro sběr dat od velké skupiny uživatelů. Není nikterak striktně specifikován na určitou funkci a tak představuje zcela univerzální řešení, kterého lze využít v řadě odvětví. Nutné je však podotknout, že je stále ve fázi vývoje a v nejbližší době očekáváme jeho první neoficiální verzi. Tato práce přispěla značnou měrou k jeho budoucí realizaci, jelikož má za úkol samotné sestavení optimalizovaného zpětného stromu.

Jednou z jeho praktických realizací představuje oblast IPTV. Hlavní funkcí TTP je tvorba, správa a zabezpečení zpětného kanálu pro RTP/RTCP relaci. U této služby existuje totiž velký předpoklad k tomu, že ji začne využívat stále větší množství uživatelů a klasický kanál RTCP přestane vyhovovat novým podmínkám, především počtu účastníků. Jak již bylo řečeno, jedná se o protokol navržený před více než 10 lety, kdy se nepočítalo, že RTCP bude u služeb s tak velkým počtem uživatelů využíván. IPTV poskytuje služby obrovské množině uživatelů, která pak ve svém důsledku produkuje obrovské množství zpráv, které je nutné přenést zpětným kanálem.

V původní specifikaci protokolu RTP/RTCP existují v podstatě jen 2 členové (vyjma speciálních mixerů) [1], kteří mezi sebou komunikují. Jedná se o Vysílač (Sender, S) a přijímač (Receiver, R). (R) je zpravidla v dnešních podmínkách připojen do určité multicastové skupiny a z ní přijímá multimediální data, které sem (S) posílá. Data jsou přenášena pomocí protokolu RTP, u něhož není kvalita služby žádným způsobem ovlivněna počtem (R). Problém však nastává u protokolu RTCP. Ten je svázán s protokolem RTP a je zodpovědný za přenos zpráv o kvalitě a stavu relace. Při malém objemu uživatelů funguje vše naprosto spolehlivě a přidělená šířka pásma je rovnoměrně dělena mezi všechny členy, kteří jsou v ní schopni přenést všechny potřebné informace v dostatečně krátkém intervalu. Když se však do relace zapojí příliš velké množství uživatelů, vzroste interval odesílání zpráv RR nad únosnou hranici a zprávy od koncových uživatelů jsou doručovány se značnou časovou prodlevou. Informace, kterou pak (S) zpracovává může být až minuty stará a v danou chvíli již nepotřebná. Z tohoto důvodu byla navržena metoda Hierarchické agregace (HA, Stromové struktury), kam byly začleněny nové prvky tak, aby společně vytvářely mechanismus umožňující provoz služby i nad obrovským množstvím uživatelů. Mezi tyto členy patří: Sumarizační uzel zpětné vazby, tzv. Feedback Target (FT), Kořenový uzel zpětné vazby, tzv. Root Feedback Target (RFT) a Manažer uzlů zpětné vazby, tzv. Feedback Target Manager (FTM). Více Obr. 2.1.



Obr. 2.1 - Hierarchická struktura zpětného kanálu

## 2.1 Způsoby zvýšení počtu příjemců

Noví členové ve zpětném RTCP kanále implementují 2 hlavní mechanismy, které nám umožňují obrovským způsobem redukovat množství dat přenášených v síti. Jedná se o sumarizaci RR zpráv koncových členů v uzlech zpětné vazby (FT) a dále pak tvorbu vyváženého hierarchického stromu (metoda hierarchické agregace, HA [7]), díky kterému jsme schopni poskytnout dostatečnou šířku pásma libovolnému počtu uživatelů v přidružené oblasti (FT). Pro zajištění maximální funkčnosti a stability v sobě protokol TTP dále implementuje monitorovací mechanismus ve vytvořeném zpětném kanále, který je nutný pro optimalizaci přenášené informace a v situacích, kdy v případě výpadku jednoho ze sumarizačních členů (FT), dokáže (FTM) aktivovat jiný (FT), který jeho roli v dostatečně krátké době převzme.

- agregace zpráv v sumarizačních uzlech (FT a RFT)
- stromová struktura sítě zpětného kanálu (metoda hierarchické agregace)

### 2.1.1 Sumarizace

Každý z příjemců generuje vlastní RR zprávu, která obsahuje informace o kvalitě příjmu paketů z multicastové skupiny. Mezi tyto informace patří mimo jiné i počet ztracených paketů (za celou dobu spojení a od poslední RR zprávy) a interval kolísání příjmu paketů (jitter). Právě tyto parametry mají největší vliv na kvalitu služby, která je v podobě IPTV nabízena. Na Obr. 2.2 je vyobrazena přesná struktura informačního paketu posílaná od příjemce k vysílači definovaná podle standardu RTCP. (Podrobné informace o jednotlivých blocích dat naleznete v dokumentu RFC 3550 [1]).

1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver	P	RC				PT = RR = 201					délka										
SSRC vysílače																					
SSRC_1 (SSRC prvního zdroje)																					
Zlomek ztráty				Celková počet ztracených paketů																	
Rozšířené sekvenční číslo																					
Kolísavost příchodu paketů																					
poslední SR (LSR)																					
Zpoždění od posledního SR (DLSR)																					

RTCP  
Hlavička

Blok zprávy

Obr. 2.2 - Struktura informačního paketu RTCP posílaného k vysílači

RR zpráva obsahuje spousty informací, které (S) nakonec vůbec nepoužije. V klasickém pojetí RTCP jsou však všechna data odesílána podle specifikace každým účastníkem zvlášť, a tak zbytečně vyčerpávají množství dat, které se zpětným kanálem smí odeslat. (S) nakonec stejně provede sumarizaci dat, aby zjistil jaké je procentuální rozložení kvality příjmu všech jeho členů. Proto je velice výhodné tuto sumarizaci provádět za něho mnohem dříve a zpětným kanálem posílat jen data, která opravdu využije. Každý člen je připojen ke svému sumarizačnímu bodu a odesílá zprávy přímo na jeho rozhraní. Sumarizační bod (FT) se stará o hlavní redukci a kompresi posílaných dat. Ve své podstatě se pro (R), kteří jsou mu podřízeni, tváří jako (S), od kterého data přijímají. Jeho prací je však získávat informace o aktuálním stavu všech svých připojených účastníků, informace z jejich RR zprávy sumarizovat pomocí histogramu a dále je odesílat již ve formě sumarizační zprávy (RSI, Receiver summary information) [2]. RSI paket se stále ještě nestal standardem v rámci dokumentů RFC a existuje zatím jen v podobě internetových draftů. V současné době je aktuální již 14ctá verze, které je volně dostupná k nahlédnutí [2].

### 2.1.2 Sumarizační paket (RSI)

Sumarizační paket (Receiver Summary Information packet, RSI) je nástrojem pro kompresi dat a umožňuje přenos dalších informací ve zpětném kanále. Je generován každým (FT), který od své skupiny příjemců obdrží zprávy o kvalitě služby. Z těchto zpráv vyjme pouze užitečnou informaci a z ní vytvoří histogram. Dochází zde tedy k zahazení všech nepotřebných hlaviček a tvoří se hlavička nová, která je několikanásobně menší. Informace uložená v samotném RSI paketu přináší další úsporu dat, jelikož struktura paketu obsahuje již připravené bloky a v daném bloku se pouze zvýší čítač o jednu, tam kde hodnota patří do definovaného intervalu. Velikost paketu je tedy konstantní a je definována množstvím parametrů a počtem uživatelů, od kterých se data získávají. Vždy totiž musíme pamatovat na situaci, že všichni uživatelé splňují danou podmínku, takže každý blok musí být schopen pojmout informaci od všech uživatelů. Obecná rovnice pro výpočet velikosti RSI vypadá následovně. Podrobnější popis velikosti sumarizačního paketu naleznete v kapitole 3.3.4.

$$PL_{RSI} = Hlavička_{RSI} + \log_2(N) \cdot \left\{ \sum (Vlastnost \cdot Počet\_intervalů) \right\} \quad [bit] \quad (2.1)$$

kde

<b><math>PL_{RSI}</math></b>	[bit]	značí velikost RSI paketu
<b>Hlavička<sub>RSI</sub></b>	[bit]	velikost hlavičky RSI paketu
<b>N</b>	[-]	počet uživatelů
<b>Vlastnost</b>	[-]	definuje parametr, který se v rámci RSI bude přenášet
<b>Počet_intervalů</b>	[-]	definuje jemnost intervalu na který je dělena určitá vlastnost

Na Obr. 2.3 a Obr. 2.4 je stručně nastíněn tvar RSI paketu. Velikost RSI paketu lze rozdělit na část statickou a část dynamickou. Statická část má pevně definované parametry. Dynamická část představuje možnost definování různých parametrů, které jsou od (S) požadovány. RSI paket ukládá záznamy ve formě histogramu. Histogram je rozdělen na řadu úseků (Obr. 2.4). Jejich velikost je závislá především na počtu uživatelů a detailnosti informace, kterou požaduje (S). Histogram není nic jiného než přenos požadované zprávy, která informuje (S), kolik uživatelů spadá do určitého intervalu. V každém intervalu musí být ponecháno místo pro zápis všech uživatelů, protože může teoreticky nastat situace, kdy všichni uživatelé splňují stejnou podmínku a spadají do jednoho intervalu.

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
SRBT={20, 21, 22}										Délka										Port																			
IP adresa																																							
úroveň ve stromu + další																																							
Vnořené části RSI (tzv. subreport bloky)																																							

Obr. 2.3 - Základní struktura sumarizačního paketu (RSI)

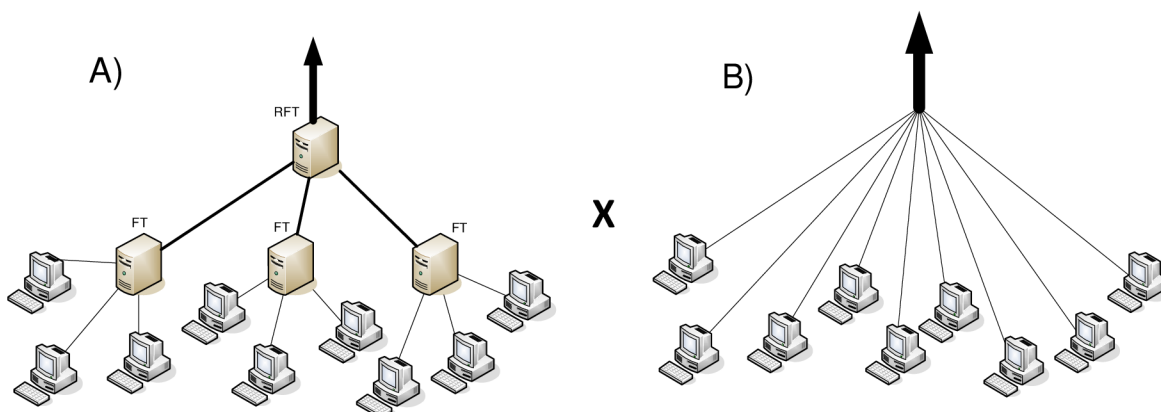
										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Histogram RTT (hlavička)																																							
Histogram data																																							
...																																							
Histogram JITTER (hlavička)																																							
Histogram data																																							
...																																							
Histogram LOSS (hlavička)																																							
Histogram data																																							
...																																							
Velikost skupiny & průměr																																							

Obr. 2.4 - Příklad struktury vnořných částí RSI

### 2.1.3 Stromová struktura (Hierarchická agregace)

Hierarchická agregace s využitím stromové struktury zpětné sítě [6] je dalším nástrojem, který nám významným způsobem pomáhá redukovat množství dat přenášející potřebné informace. Stromová struktura zavádí do zpětného kanálu určitou disciplínu. Pro vysílač je životně důležité, aby věděl, jakým způsobem jsou data, která vysílá, přijímána, a jaké mají vlastnosti po doručení. Kritickou vlastností je pak časová organizace. Jednoduše řečeno data, která nám přijdou s příliš velkým zpožděním, pro nás v danou chvíli nemusí mít již téměř žádnou vypovídající hodnotu. Pokud se vysílač dozví o problémech u příjemců skrz RR zprávy, je schopen na ně reagovat nejrůznějšími způsoby. Může změnit kódování přenášených dat, aby byla data na cestě lépe zabezpečena, anebo může změnit formát obrazu, videa či zvuku. Pro (R) je vždy mnohem schůdnějším řešením obraz či zvuk s horší kvalitou, než špatná plynulost přehrávání, která přináší rozhodně větší rušivý vjem.

Stromovou strukturu musí samozřejmě někdo sestavit a dále jí i monitorovat a svým způsobem řídit. Tímto článkem neboli „mozkem“ celého protokolu je Feedback Target Manager (FTM), jehož funkce budou popsány v kapitole 2.4.3. Pro nás je důležité, aby stromová struktura byla za každých podmínek maximálně vyvážená a aby byl každý sumarizační bod optimálně a efektivně vytížen. Totiž, čím více vrstev hierarchický strom obsahuje, tím větší je výsledné zpoždění přenosu feedback zprávy a tím se tato zpráva více znehodnocuje. Existují však jisté metody, které budou v této práci zmíněny, jak vliv zpoždění přenosu zprávy eliminovat a poskytnout tak maximální možnou kvalitu služby za daných podmínek. Na Obr. 2.5 je znázorněn princip celého protokolu. Klasický způsob přenosu zpráv ve zpětném kanále produkuje mnohem větší šířku pásma potřebnou pro přenos stejné informace v porovnání s protokolem TTP. Jak je názorně vidět, pomocí sumarizace zpráv a správné vyváženosti hierarchického stromu lze dosáhnout velké úspory množství dat.



Obr. 2.5 - Datový tok s hierarchickou agregací (A) a bez ní (B)



Hlavní rozdíl mezi oběma metodami představuje množství dat, které je doručeno k (S). Na Obr. 2.5B je šířka pásma zpětného kanálu ( $BW_{RR}$ ) rovna následující rovnici:

$$BW_{RR} = \frac{PL_{RR} \cdot N}{T_{RR}} \quad \left[ \frac{\text{bit}}{\text{s}} \right] \quad (2.2)$$

a na Obr. 2.5A za použití Rovnice 3.11, která je popsána v kapitole 0, je šířka pásma zpětného kanálu ( $BW_{SR}$ ) rovna následujícímu vztahu:

$$BW_{RR} = \frac{PL_{RSI}}{T_H} \quad \left[ \frac{\text{bit}}{\text{s}} \right] \quad (2.3)$$

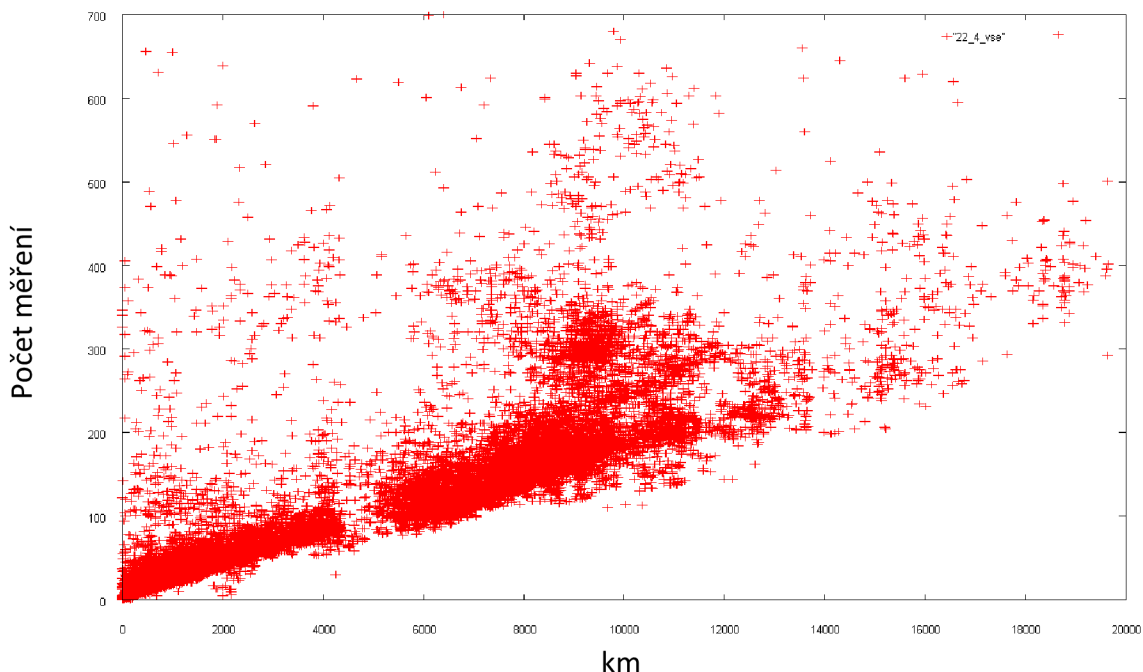
kde

$PL_{RSI}$	[bit]	značí velikost RSI paketu
$T_H$	[s]	interval odeslání RSI paketu

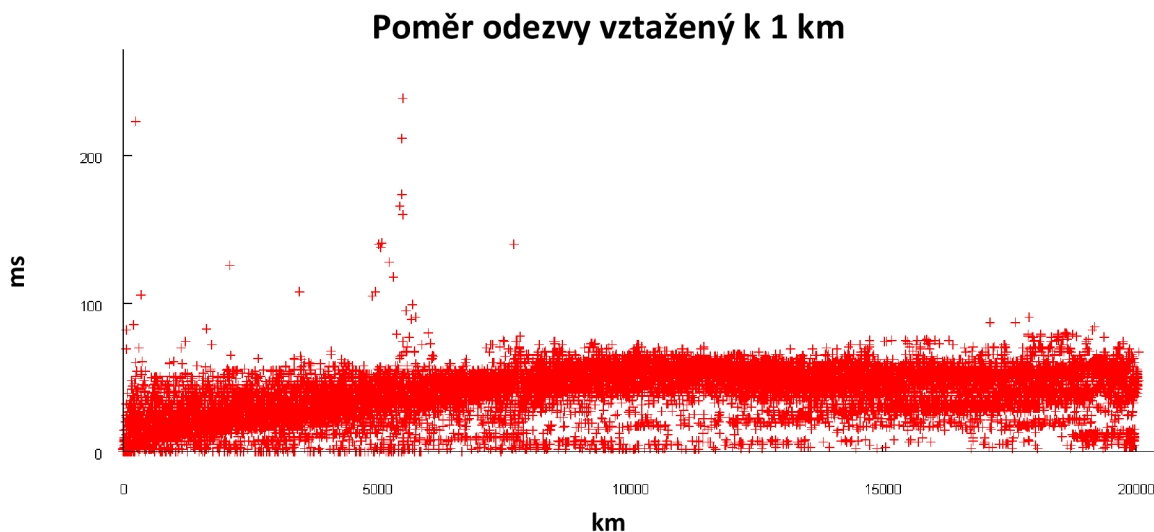
## 2.2 Zajištění lokalizace příjemce

K tomu, aby sem mohla ustanovit správná stromová struktura a pakety mohly být doručovány s maximální rychlostí, je třeba znát lokalizaci jednotlivých uživatelů. Pokud budeme předpokládat existenci sumarizačních uzlů, je nesmysl, aby se uživatel nacházející se momentálně v Evropě přihlašoval k sumarizačnímu uzlu, který se fyzicky nachází v Americe. Pro zjištění pozice jednotlivých bodů existuje řada metod založených na několika principech. Zpravidla se ale jedná o přepočítání časové odezvy pevně určených bodů v síti, tzv. LandMark serverů (servery pozičních bodů) [3]. Poziční bod je zařízení, o které víme, kde se nachází a jakou má pozici určenou pozičním vektorem. Na světě může existovat celá řada takovýchto serverů, s jejichž pomocí jsme následně schopni vypočítat umístění uživatele kdekoli na Zemi. Celý princip je vlastně totožný se systémem GPS až na to, že pro výpočet polohy přijímače neslouží informace cyklicky vysílaná družicí, ale je generována na základě metody RTT (Round Trip Time). K tomuto měření se používá protokol ICMP [8] realizovaný příkazem ping. Uživatel provede asi 10 dotazů na vzdálený server, aby zjistili průměrnou odezvu a tím i přibližnou vzdálenost. I když je ve skutečnosti vypočtená vzdálenost získána pro cestu na spoji, rozdíl oproti skutečné vzdálenosti je pro naše použití dostačující. Na toto téma provedla kolegyně Lucie Jarošová měření a výsledek její studie naleznete na Obr. 2.6 a Obr. 2.7.

**Závislost doby odezvy na vzdálenosti uzlu v síti**



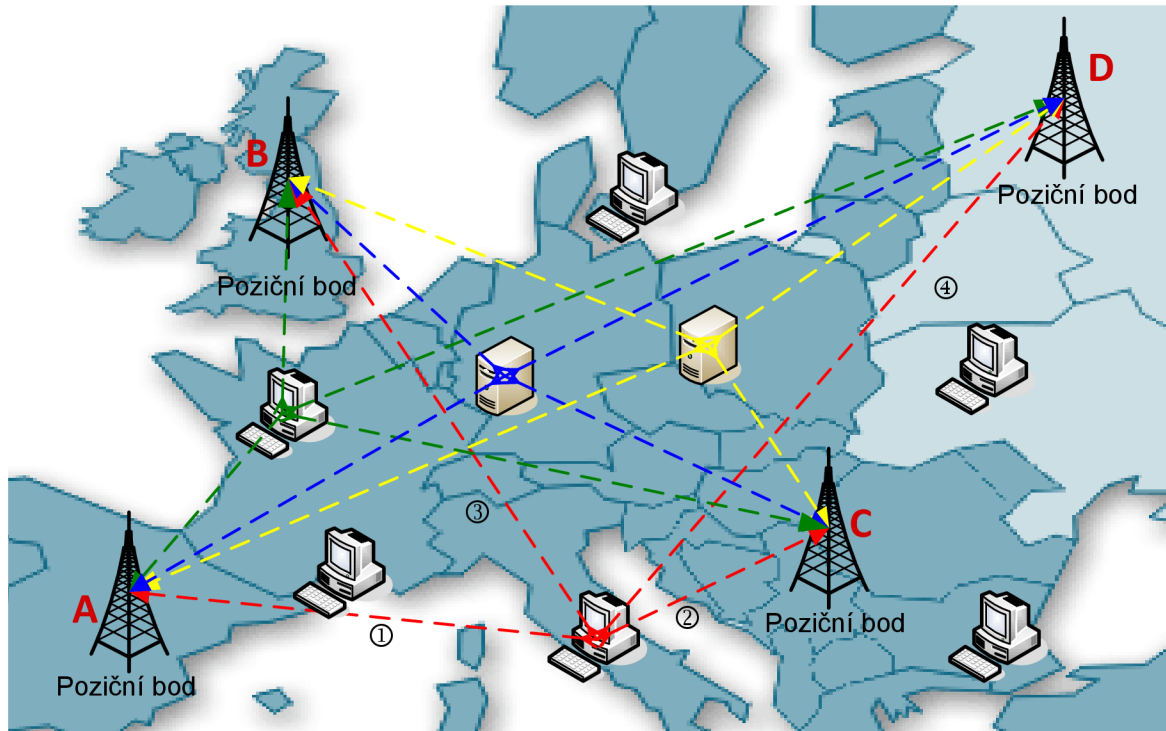
**Obr. 2.6 - Doba odezvy ke vzdálenému uzlu v síti**



**Obr. 2.7 - Poměr odezvy**

Z obou grafů je patrné, že doba odezvy jsou při tomto druhu měření velice stabilní a závislé právě na vzdálenosti uzlu. První graf představuje celkový čas odezvy od uzlu vzdáleného v km v síti a na druhém grafu je vyobrazena změna odezvy v poměru na 1 km. I zde je vidět, že téměř nezáleží na vzdálenosti uzlu v síti a poměr zůstává téměř konstantní.

Zajištění lokalizace však může v určitých stavech způsobit značné nepříjemnosti, na které by se nemělo zapomínat. Pokud bychom předpokládali současně připojení 1 000 000 uživatelů, může počet dotazů na poziční bod způsobit jeho nedostupnost, tzv. DoS útok. V reálných podmínkách tedy musíme zabezpečit, aby se současně v jeden okamžik nemohlo přihlašovat k jednomu pozičnímu bodu příliš velké množství uživatelů. Podle studie provedené ve článku [3] dokáže obyčejný počítač s procesorem 1,8GHz obsloužit až 2500 dotazů za vteřinu, tudíž je třeba do systému dotazování vložit jistou náhodnost při výběru pozičního bodu. Dále lze pak pro posílení funkčnosti využít procesu clusterování serverů. Na Obr. 2.8 jsou znázorněny 2 PC a 2 uzly v procesu vytváření tabulek vektorů.  $PC_1$  se nachází někde v Itálii. RTT dotazem na poziční body A, B, C a D získá  $PC_1$  tabulku vektoru, jaká je naznačena v Tab. 1, a výsledek zašlou (FTM). Ten si podle těchto informací aktualizuje svůj seznam členů, a pokud zjistí, že se v některé z oblastí nachází již příliš mnoho zařízení obsluhovaných jedním sumarizačním bodem, dokáže zajistit aktivaci nového uzlu, který pomůže s datovou reží. Pro vypočtení pozice stačí znalost vzdálenosti od 3 pozičních bodů, ale pro přesnější hodnoty je dobré využít většího počtu pozičních uzlů.



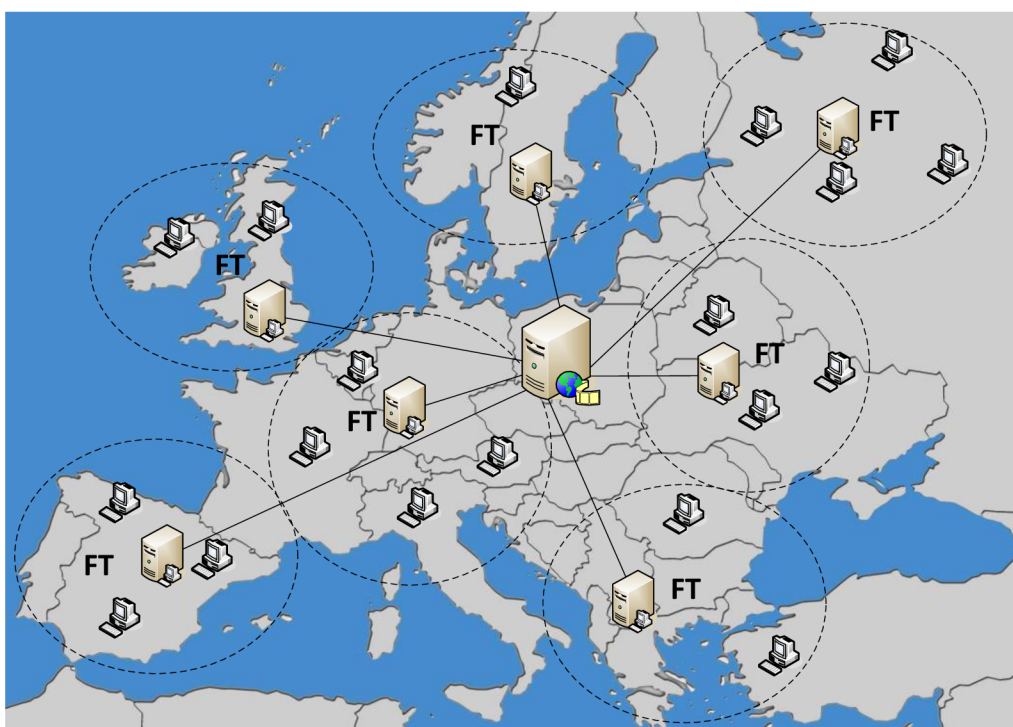
Obr. 2.8 – Vektory pozičních bodů

Tab. 1 - seznam vektorů

	IP pozičního bodu	Jméno	Průměrná odpověď (ms)
①	132.122.112.2	A	85
②	132.122.112.3	B	79
③	132.122.112.4	C	120
④	132.122.112.5	D	250
⑤	...	...	...

## 2.3 Příslušnost k sumarizačnímu bodu

Proto, aby mohla být RR zpráva doručena až k (S) tím nejrychlejším způsobem, musí znát (R) polohu nejbližšího sumarizačního bodu (FT). Seznam sumarizačních bodů je k (R) vysílán pomocí multicastového SR kanálu (více kapitola 2.4.3 a 2.5.1) a z těchto informací si (R) sám vypočte nejbližší (FT), ke kterému bude odesílat své pakety. Výpočet je prováděn z jednoduchého porovnání vlastních vektorů pozicních bodů s vektory, které přijal v SR kanálu. Množství a stav sumarizačních bodů spravuje (FTM) a informuje o změně všechny přijímače pomocí multicastu. Proces nalezení příslušnosti k určitému uzlu bývá odborně nazván jako „binning“ [3]. Jak již bylo popsáno v kapitole 2.2, každý klient kontaktuje definované pozicní body, aby si pomocí ní vytvořil vlastní tabulku vektorů. V případě, že vzniknou nějaké potíže s dostupností určitého serveru, tyto problémy řeší (FTM) (více dále). Náhodné pořadí je pouze a jen z důvodu rozptýlení dotazů na servery, protože pokud by se v jednu chvíli přihlásilo příliš velké množství uživatelů, mohlo by dojít k naprostému zahlcení služby pozicního bodu. Taková situace může nepochybně nastat například v případě, kdy je vysílán nějaký oblíbený program. Nejmenší rozdíl odchylky ve výpočtu vektorů od daného (FT) je vyhodnocen jako pozicně nejbližší. Jak je vidět na Obr. 2.9, klienti jsou po procesu „binningu“ organizováni do skupin a těmto skupinám se říká „bins“ („koše“). Pro naše použití mají „koše“ význam skupiny uživatelů patřící do území jednoho (FT). Každý „koš“ má své omezení počtu klientů a toto omezení je závislé na možnostech daného sumarizačního uzlu (FT). Jak již bylo zmíněno výše, zajištění těchto vlastností sleduje (FTM), který je v případě potřeby schopen vytvořit nové sumarizační uzly (FT), popřípadě je i rušit a donutit tím uživatele, odesílat zprávy na jiný (FT).



Obr. 2.9 - Proces vytváření "košů"

## 2.4 Členové zpětného kanálu

V této sekci budou popsáni jednotliví členové TTP relace a jejich případná rozdílnost funkce s porovnáním funkce u relace používající klasický zpětný kanál protokolu RTCP.

### 2.4.1 Vysílač (S) a příjemce (R)

Pozice a účel těchto dvou členů relace je naprosto zřejmý. V rámci hierarchické agregace se pro oba prvky nic zásadního nemění. (S) stále odesílá multimediální data do multicastové skupiny a (R) tvoří informace o kvalitě příjmu těchto dat. Jedinou změnu pro (S) představuje přeposílání dat od (FTM), ve kterých je obsažena informace potřebná pro hladký provoz celé stromové struktury. (S) tyto data přijme již zpracovaná na svém vstupním rozhraní a pouze je přepošle do SR kanálu, aby si je mohli všichni členové relace přečíst. Pro (R) je situace také víceméně stále stejná až na pár odlišností. (R) čte data z multicastové skupiny a podle definovaného parametru zpoždění, následně chybovosti příjmu dat a dalších, generuje RR zprávy. Až doposud se nic nemění. Rozdíl však nastává při odeslání dané RR zprávy. Za normální situace, by odeslal paket přímo (S), aby ho informoval o kvalitě svého příjmu, ale v relaci s podporou TTP protokolu odesílá zprávu k nejbližšímu (FT), který simuluje stav (S) a sbírá informace od všech účastníků, kteří patří do jeho skupiny. O tom, kterému z řady (FT) odešle (R) svoji zprávu rozhodne sám (R) na základě výpočtu své pozice. Následně může z dat přijímaných z SR kanálu zjistit pozici nejbližšího dostupného (FT). Na zjištění globální pozice existuje řada studií a metod. Mezi neznámější patří Global Network Positioning (GNP)[9] a Vivaldi [10]. Pokud tedy shrneme rozdíl působení (R) v topologii oproti standardu RTP/RTCP, tak jedinou zásadní změnu představuje zjištění pozice nejbližšího (FT) pomocí informací zasílaných v SR kanálu. Zjištění pozice je možno provádět několika způsoby o čemž pojednávají články [9], [10] a [11].

### **2.4.2 Sumarizační uzel (FT) a Kořenový uzel (RFT) zpětného kanálu**

Asi největší inovaci a především mechanismus, který umožňuje realizace celého protokolu, představuje právě (FT). (FT) je zodpovědný za příjem zpráv od (R), pro které představuje svým způsobem (S). (FT) má vždy představu o tom, kolik uživatelů je k němu připojeno a jaký interval odezvy jim poskytuje. Tyto informace načítá z kanálu SR. Jeho hlavní činnost totiž spočívá ve shromažďování informací v časovém intervalu (obecně 5s) a odesláním sumarizované informace nadřazenému (FT). V časovém intervalu, kdy mu všichni jeho členovou odešlou zprávu, vypočítá tzv. sumarizační paket (RSI, Obr. 2.3). V sumarizaci totiž tkví ona datová úspora a možnost rozšíření služby na několika násobně větší počet zákazníků. (FT) poté odešle vytvořený RSI paket nadřazenému (FT), který provede další sumarizaci přenášené informace. Tento proces pokračuje dále, dokud není zpráva přijatá RFT, který představuje koncový sumarizační článek celého stromu. Pro větší představu a názornost zpět na Obr. 2.1. Sumarizace RR popřípadě RSI zpráv nepředstavuje nic jiného než ukládání informací do histogramu. (S) nepotřebuje znát data od každého (R) jednotlivě. Pro něho je důležité procentuální vyjádření kvality příjmu všech uživatelů. Ve své podstatě by stejně tuto sumarizaci musel provést sám a proto je tvorba histogramu v síti ideální nástroj ke snížení datového toku informací a tím i ušetření velké šířky pásma. (S) může na základě těchto informací změnit formát nebo samotné kódování dat, ale tato změna vždy ovlivní celou skupinu, proto musí být prováděna, jen v případě, kdy z toho bude těžit většina členů. Individuální potíže s příjmem dat pak musí vyřešit každý člen sám, jelikož s největší pravděpodobností bude problém spíše s jeho připojením.

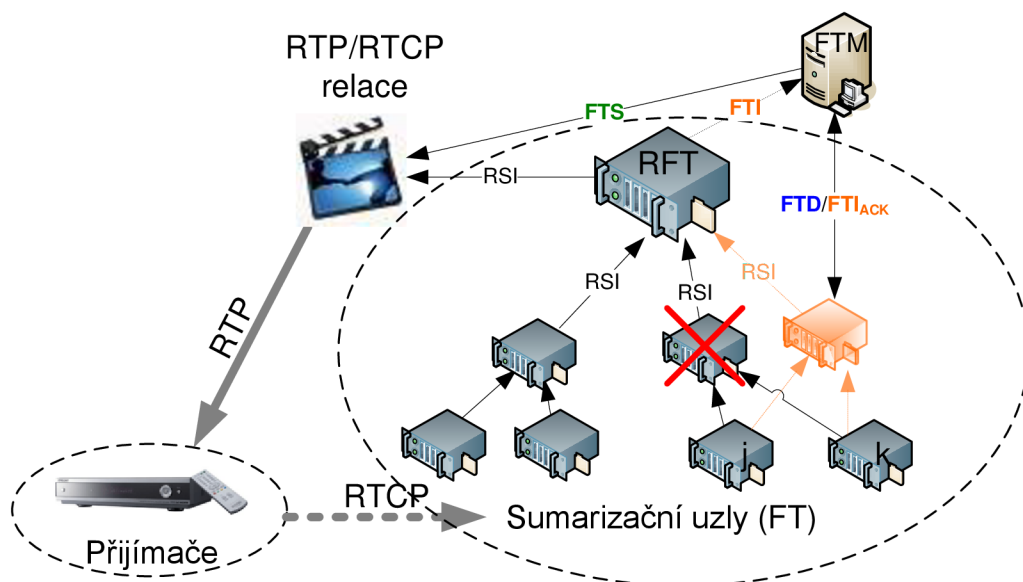
### **2.4.3 Manažer uzlů zpětné vazby (FTM)**

Pomyslným „mozkem“ celého protokolu je právě Manažer uzlů zpětné vazby (FTM). Jeho roli lze v rámci celého systému popsat jako monitorující a částečně řídicí. Jedná se o jediný prvek v síti, který na sebe přebírá centrální správu. Přesto by i po jeho výpadku měla být síť dále provozuschopná a uživatel by neměl zaznamenat po určitý čas žádné potíže. (FTM) totiž nastaví parametry celého stromu společně s rezervními stavy, které zaručují v případě velkého nárůstu uživatelů a zároveň momentální nefunkčnosti (FTM) neustály běh systému.

Práce (FTM) spočívá v sestavení hierarchického stromu, následný dohled nad jeho funkčností a s tím spojenou optimalizací. (FTM) přijímá zprávy jak od (FT), které jsou mu nepřímo podřízeny tak od (S), aby dokázal optimálně reagovat na všechny stavy stromu. Jeho funkce při řízení provozu ve stromu není přímá, ale představuje jakousi vzdálenou zprávu nad celou topologií. Hierarchický strom je schopen pracovat určitý čas (kdy nedochází k velkým výkyvům počtu připojených (R)) bez nutnosti spolupráce s (FTM). Ono se ani nepředpokládá, že by se stav stromu měnil ve velmi krátkých intervalech, ba naopak. Abychom dosáhli co možná nejlepších výsledků, je třeba zajistit plynulý provoz a

setrvačnou (dynamickou) změnou topologie stromu. V praxi to ve většině případů bude vypadat asi tak, že v průběhu pořadu zůstane strom neměnný a ke změně dojde v období, kdy program končí, či začíná nový. Vše závisí především na sledovanosti a oblíbenosti daného pořadu. (FTM) tedy představuje aktivní prvek, který pasivně (rozesíláním informace do sítě) ovlivňuje stav a chování (R) nebo (FT). Proces optimalizace hierarchického stromu, který je hlavní náplní práce (FTM), je probírán v kapitole 3.3.

Na Obr. 2.10 je ve stručnosti popsána práce (FTM) při výpadku jednoho ze sumarizačních uzlů. (FTM) si udržuje informace o všech prvcích v síti a pokud dojde k nějaké neočekávané události v síti, je o této události informován. Na základě přijaté informace (FTM) zjišťuje, co je třeba provést za změny v síti. V rámci protokolu TTP existují typy paketů, které dokážou aktivovat (deaktivovat) nový (FT) nebo nastavit nové parametry aktivních (FT). Tyto typy paketů budou probrány v následující kapitole 2.5. Pokud tedy dojde v síti k výpadku jednoho ze sumarizačních bodů (FT), (FTM) dostane od nadřazeného (FT) informaci o změně. (FTM) reaguje takový způsobem, že vyhledá nejvhodnější nový (FT) ve svém seznamu a pokusí se ho aktivovat. Po úspěšné aktivaci pak rozešle aktualizované informace pomocí multicastového SR kanálu. Sumarizační body (FT) číslo 1 a 2 v Obr. 2.10 se dozvědí o nedostupnosti svého nadřazeného uzlu z SR kanálu a vyhledají si nový (FT). Tím se pro ně stane právě nově aktivovaný (FT) a své zprávy začnou vysílat k němu. Význam a funkce jednotlivých paketů bude objasněn v kapitole 2.5 a celé kapitole 4.



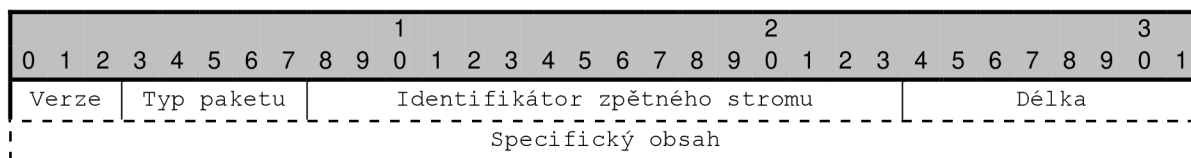
Obr. 2.10 - Práce (FTM) při výpadku sumarizačního uzlu



## 2.5 Pakety pro sestavení a správu zpětného kanálu

Aby bylo umožněno fungování celé hierarchické struktury při sběru dat, bylo třeba vyvinout nový protokol, který by se o tuto činnost staral. Protokol TTP je projektován tak, aby byl schopen pracovat samostatně a definoval vyvážený strom pro sběr informací od velké množiny uživatelů. Pokud se zaměříme čistě na IPTV, TTP protokol doplňuje mezery a omezení multimediálních služeb, které jsou citlivé na zpoždění a délku intervalu doručení zprávy o kvalitě poskytované služby a které jsou určeny pro velké množství koncových zákazníků. Ke své činnosti v rámci IPTV potřebuje protokol TTP „symbiózu“ s protokolem RTP/RTCP.

V původním návrhu [5] je definováno celkem 5 druhů paketu. Od té doby prošel protokol řadou změn, a mnohokrát ho bylo třeba přepracovat, aby odpovídal všem požadavkům. Jak již bylo řečeno dříve, protokol je stále ve fázi vývoje a v momentální verzi 0.5.0 jsou prozatím definovány 3 typy zpráv (FTS,FTD,FTI), které dohromady tvoří komplexní množinu pro budování a dynamickou zprávu parametrů hierarchického stromu. Redukce počtu paketů je dalším z mých přínosů, která pomohla zjednodušit návrh a realizaci. Hlavička TTP paketů je znázorněna na Obr. 2.11.

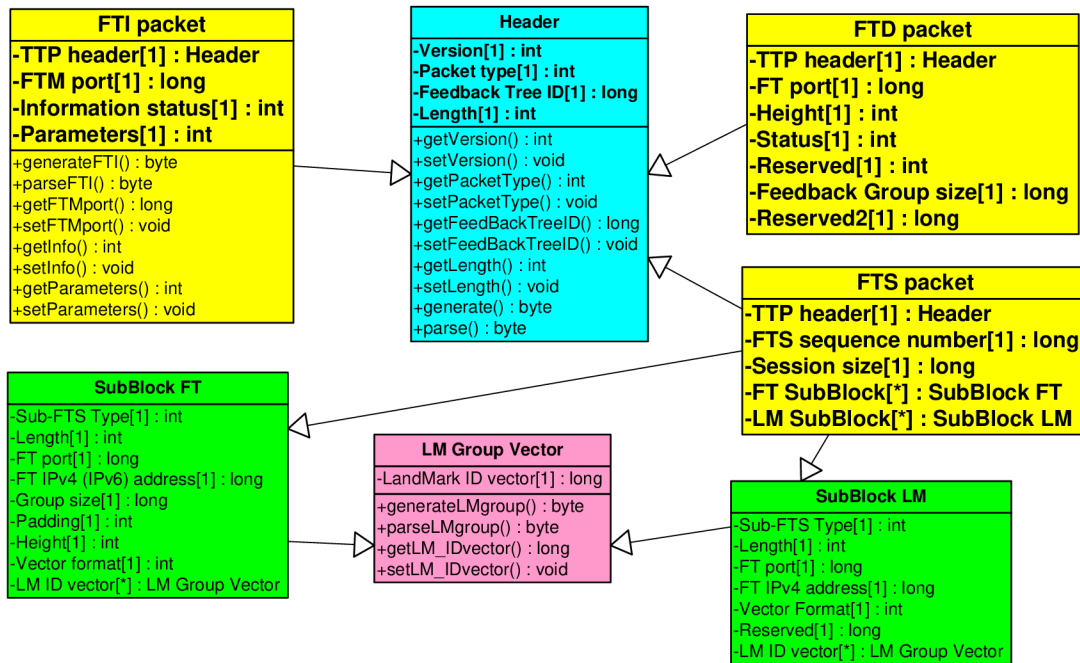


Obr. 2.11 - Hlavička všech paketů TTP

- **Verze** - 3bity určují verzi paketu
- **Typ paketu** - 4bity připraveny pro typ paketu,
- **Identifikátor zpětného stromu** - 16bitů pro ID stromové struktury (jelikož ve stejném čase může být sestaveno více IPTV relací)
- **Délka** - 8 bitů definuje délku zprávy, udává velikost v počtu 32bit slov (samotná hlavička se do velikosti nepočítá)

Zbytek paketu je již specifický pro různé druhy paketů TTP protokolu. Následující část bude věnována stručnému popisu jednotlivých paketů, aby se tento text nestal nepřehledným. Na přiloženém CD však naleznete kompletní zdrojový kód, v kterém lze najít detailní popis jednotlivých částí paketů společně s jejich metodami. Nemělo by smysl v tomto textu nějak detailně popisovat jednotlivé vlastnosti každého paketu, jelikož se stále nejedná o konečnou verzi, ale o návrh, který se v závislostech na zjištěné skutečnosti bude měnit. Na Obr. 2.12 naleznete UML [16] diagram, který shrnuje celou strukturu TTP paketů, jak jsou tvořeny a jaké atributy a metody obsahují. Byl použit zjednodušený výpis,

kdy jsou metody naznačeny pouze u třídy „Header“ a „FTI packet“, které za třídy „Header“ dědí. Atributy paketů jsou vypsány u všech paketů a chybějící metody ostatních tříd jsou analogické k uvedeným. Kompletní UML diagram tříd lze nalézt v dodatku na konci práce a pochopitelně na přiloženém CD.

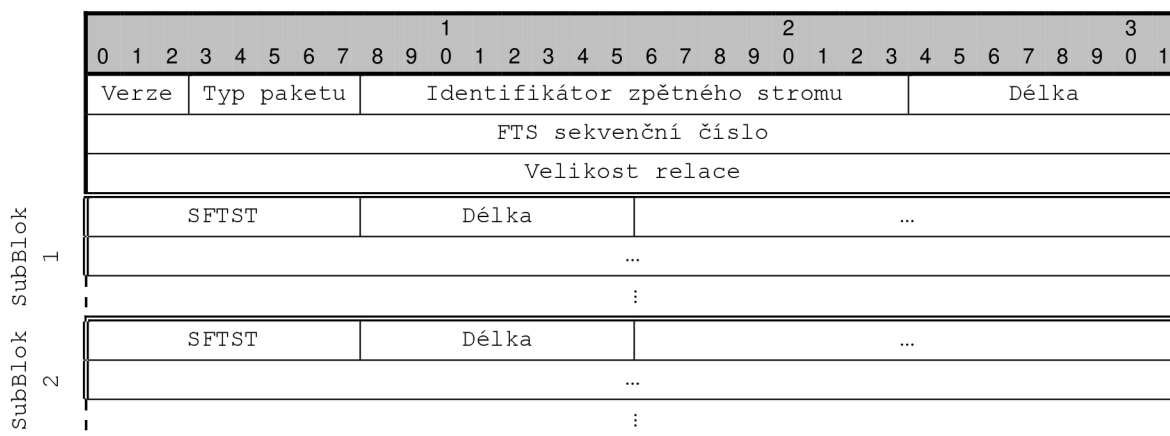


Obr. 2.12 - UML diagram tříd

## 2.5.1 Specifikační paket zpětné cesty (FTS)

Paket FTS (Feedback Target Specification packet) přenáší hlavní informace o tom, jak vypadá aktuální hierarchický strom a které členy se tohoto stromu účastní. FTS paket je generován (FTM) a odesílán na rozhraní (S), který ho přímo přeposílá do kanálu SR. Množství dat je závislé na velikosti hierarchického stromu a na způsobu, jakým (FTM) tuto informaci sděluje. Pokud bychom spravovali velký strom o několika vrstvách, poskytující rychlou odezvu, je pro tuto realizaci třeba obrovské množství (FT), jak bude rozebráno v kapitole 3. Informace o aktuálním stavu stromu musí být rychle a jednoduše dostupná všem (R). Ať je velikost FTS paketu jakákoliv, nesmíme dopustit, abychom touto zprávou celý kanál SR zahltili. Proto je nastaveno cyklické odesílání FTS s informací o celé síti po určitých intervalech (standardně 5s) a pokud dojde ke změně v rozmezí tohoto intervalu, posílá se již jen rozdílová FTS zpráva pouze o (FT), kteří změnili svoji činnost. Tím je zaručeno, že každý člen je schopen načíst informaci o stavu zpětného kanálu téměř okamžitě. Rozdílové zprávy FTS o měnících se (FT) pak pokrývají rychlé přechody stavů hierarchického stromu.

V každém FTS se nachází několik tzv. SubBloků, které nesou informace o daném (FT). SubBlok může být buď typu FT (Feedback Target) nebo LM (LandMark). SubBlok-FT definuje vlastnosti (FT) a SubBlok-LM definuje vlastnosti Poziční bodu. Každý SubBlok zabírá 6\*32bit (192bitů). Zobecněná struktura paketu FTS je na Obr. 2.13.



Obr. 2.13 - Feedback target specification packet (FTS)

- **FTS sekvenční číslo** – 32bitů pro označení sekvenčního čísla zprávy
- **Velikost relace** – 32bitů pro určení velikosti celé relace  $2^{32}=4,3$ miliardy příjemců, což by teoreticky mělo stačit na IPTV jakéhokoliv rozsahu
- **SFTST** – značí typ paketu, který se v rámci SubBloku posílá

### 2.5.1.1 SubBlok-FT

SubBlok definující vlastnosti (FT) má v TTP relaci jednu z největších informačních hodnot. Díky této informaci jsou všichni členové schopni získat přehled, který z (FT) je vlastně aktivní a kde se nachází. (R) totiž potřebují, jak už bylo několikrát poznamenáno, najít ten nejbližší možný (FT), aby doručení jejich RR zpráv bylo co nejrychlejší. Obsahem SubBloku-FT je řada atributů, které jsou znázorněny na Obr. 2.14. Jedná se především o IP adresu aktivního (FT) a port, na kterém službu poskytuje, o definovaný počet uživatelů, který je schopen pojmout. Dále pak výšku stromu, ve které se virtuálně nachází a soubor vektorů s popisem jejich typu, aby byl (R) schopen určit pod jaké metody má spočítat pozici (FT).

SubBlock FT
-Sub-FTS Type[1] : int
-Length[1] : int
-FT port[1] : long
-FT IPv4 (IPv6) address[1] : long
-Group size[1] : long
-Padding[1] : int
-Height[1] : int
-Vector format[1] : int
-LM ID vector[*] : LM Group Vector

Obr. 2.14 - SubBlok FT

Metod k tomu, jak určit pozici uživatel, existuje celá řada. Mezi nejznámější a nejvíce používané patří zcela jistě GNP [9] a Vivaldi [10]. Metoda GNP i Vivaldi jsou velice podobné systému GPS [17], kdy uživatel podle získaných vektorů pozice pevných bodů dokáže vypočítat svojí vzdálenost k nim. Na základě RTT změří přibližnou vzdálenost od definovaných pozičních bodů a spočítá si vlastní pozici. Jedná se o proces triangulace [18] a provádí se trigonometrickým výpočtem. Sestrojí se pomyslný trojúhelník, jehož jedna strana, je strana již známého jiného trojúhelníku s dvěma koncovými referenčními body a třetím bodem je místo, jehož souřadnice se zjišťuje.

### 2.5.1.2 SubBlok-LM

Druhým z řady SubBloků je SubBlok-LM definující vlastnosti pozičních bodů. Ten to druh informace je potřebný v první fázi přihlášení uživatele do IPTV relace. Díky této informaci ví, na které IP adrese a portu se nachází server, který slouží pro účel zjištění pozice. Pro vlastnosti vektorů a jejich typu platí opět možnost použití libovolného mechanismu. Výpočet pozice není vázán na jedinou metodu. Do budoucna lze implementovat i další metody.

SubBlock LM
-Sub-FTS Type[1] : int
-Length[1] : int
-FT port[1] : long
-FT IPv4 address[1] : long
-Vector Format[1] : int
-Reserved[1] : long
-LM ID vector[*] : LM Group Vector

Obr. 2.15 - SubBlok LM

## 2.5.2 Definiční paket (FTD)

FTD (Feedback Target Definition packet) je paket, který je generován (FTM) a slouží k definování vlastností (FT). Tato zpráva je posílána výhradně unicastem přímo na adresu daného (FT). Důvod, proč je FTD posílán unicastem a ne v multicastu jako v případě FTS spočívá hlavně v rychlosti odezvy. (FT) potřebuje po příjmu aktivační FTD zprávy určitou reakční dobu k tomu, aby se „přepnul“ do aktivního stavu definovaného právě v FTD. Potvrzení či odmítnutí stavu odešle zpět k (FTM), pomocí FTI, který bude rozebrán v následující části. (FTM) mezitím informuje (R) o pozičních bodech, aby si mohly pomocí nich zjistit svoji pozici. Obecná struktura FTD paketu je na Obr. 2.16. Je zde definováno několik již pevně určených parametrů a ponecháno místo na další rozšíření. Ze základních parametrů bych upozornil na pole „Výška“, kde (FTM) odesílá (FT) informaci o tom v jaké vrstvě zpětného kanálu bude pracovat a pole „Vlastnost“, kde definuje, zda daný (FT) bude aktivním členem zpětného kanálu nebo bude poskytovat službu pozičního bodu. Dalším zajímavým blokem je pole „Velikost přiřazení skupiny“, kde je (FT) informován o tom, pro jaký počet příjemců je v aktuálním stromu postaven. Pokud by došlo k překročení tohoto limitu směrem nahoru či dolů. (FT) informuje o tomto stavu (FTM) ve zprávě FTI. Každý (FT) má totiž určený průměrný počet členů, od kterých může přijímat zprávy. Na druhou stranu má však definovanou určitou rezervu, aby mohl pojmout měnící se počet uživatel. Pokud dojde k překročení i této hranice, (FT) odesílá zprávu FTI, v které definuje vzniklou potíže, kterou (FTM) musí řešit. Více dále.

1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Verze		Typ paketu			Identifikátor zpětného stromu															Délka											
FT port										Výška					Vlastnost					Reservováno											
Velikost přiřazené skupiny										Reservováno																					

Obr. 2.16 – Definiční paket uzlu (FTD)

### 2.5.3 Informační paket (FTI)

Tato zpráva (Feedback Target Information packet) je poslední z řady a je posílána především (FT). Využití nalézá při potvrzení o přechodu do stavu, který (FTM) požadoval, anebo při situaci, kdy (FT) překračuje meze určitého parametru definovaného (FTM), například počet uživatelů ve skupině. Další možností je pak informace o nedostupnosti aktivovaného (FT), což zjistí (FT) jemu nadřizený, pokud přestanou od problematického (FT) docházet RSI pakety. Paket FTI je tedy určen pro specifické informování o stavech, které jsou definovány (FTM) a měli by se v jednotlivých uzlech dodržovat. Obecná struktura paketu je naznačena na Obr. 2.17

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Verze		Typ paketu				Identifikátor zpětného stromu																Délka																	
FTM port										Druh informace										Parametry																			

Obr. 2.17 – Informační paket (FTI)

Nejzajímavější a nejdůležitější vlastností paketu je pole s názvem „Druh informace“. Zde se přenáší informace o tom zda FTI paket potvrzuje nebo zamítá žádost (FTM) anebo informace o tom, že vlastnosti přidělené danému FT již neodpovídají aktuální skutečnosti a je třeba změnit stavbu stromu. O tom však více v kapitole 4.

### 3 SESTAVENÍ HIERARCHICKÉHO ZPĚTNÉHO KANÁLU PRO IPTV

Pro sestavení zpětné cesty od (R) k (S) existuje mnoho možností. Vždy je důležité vybrat takovou, která v daný okamžik poskytuje optimální způsob zajištění kvalitní služby. Tato práce sice pojednává o protokolu jako celku, avšak hlavní část je věnována právě sestavení optimalizovaného zpětného kanálu. Tato kapitola popisuje hlavní část protokolu RTP a to samotné jádro mechanismu tvorby hierarchického stromu. Na jedné straně máme totiž přání zákazníka, který požaduje vždy co možná nejlepší kvalitu služby a na druhé straně poskytovatele, který se snaží vyhovět přání zákazníka za co možná nejnižší konečnou cenu. Protokol RTP se snaží obě tyto rozdílné představy co nejvíce přiblížit. Zákazník má možnost využívat velice kvalitní službu a přitom náklady na implementaci pro poskytovatele nikterak závratně nerostou ani v případech, kdy chce získat informace od více než milionu koncových klientů. Samotný protokol RTP poskytuje řadu variací, podle níž lze nastavit službu s takovými parametry, které budou vyhovovat oběma stranám současně.

#### 3.1 Matematický aparát

Hlavní potíží protokolu RTP je jeho nehospodárnost s přidělenou šířkou pásma. Nejde ani tak o to, jaké množství dat je posíláno od příjemce (R) k vysílači informace (S), ale hlavně o zbytečné plýtvání šířky pásma při realizaci tohoto přenosu. Protokol RTP je navržen tak, že každý z účastníků relace posílá unicastem svá data přímo k (S). Tím pádem se zpětným kanálem odesílají nejen důležitá data, ale i informace o nepotřebných hlavičkách RR paketů, které (S) nikterak nevyužije a zbytečně zabírají potřebnou kapacitu. V rámci služby IPTV se předpokládá účast několika tisíců až milionů uživatelů a proto pro takové množství uživatelů musíme připravit potřebnou šířku pásma ve zpětném kanále, abychom zajistili dostatečně rychlé odesílání RR zpráv k (S).

Koncoví uživatelé (R) posílají svojí RR zprávu svému nejbližšímu (FT) v přístupové vrstvě hierarchického stromu. Standard RTP je definován tak, že odesílá RR zpráv v intervalu  $<5;\infty$ ) sekund. Pokud tedy vezmeme v úvahu ten vůbec nejpříznivější případ, je informace od všech (R) odeslána během jednoho intervalu 5 sekund. V rámci zpětného kanálu musí tedy každý (FT) počkat minimálně 5 sekund a teprve pak může vygenerovat RR paket a odeslat ho nadřazenému (FT). Celkové zpoždění průchodu každé RR zprávy se tedy rovná minimálně počtu vrstev provozovaného stromu násobených intervalem odesílání dat od (R).



$$T_C = H \cdot T_{RR} + \sum_1^H T_Z \quad [s] \quad (3.1)$$

kde

<b>T<sub>C</sub></b>	[s]	celkový čas průchodu zprávy přes hierarchický strom
<b>T<sub>RR</sub></b>	[s]	interval generování RR zprávy
<b>T<sub>Z</sub></b>	[s]	doba zpracování informace a generování RSI v (FT)
<b>H</b>	[-]	výška hierarchického stromu zpětné vazby

Poznámka: doba zpracování  $T_Z$  je natolik malá, že jí lze bez komplikací zanedbat

Toto zpoždění může tedy s vysokým počtem vrstev nabývat kritických hodnot a pro kvalitní fungování IPTV začne představovat problém. Pokud by se data na druhou stranu odesílaly rychleji, docházelo by pak k příliš velkému vytížení kanálu, což by v konečném důsledku přineslo jen komplikace a žádnou výraznou výhodu. Aby byl přenos zpráv od (R) přes zpětný kanál efektivně využit, musíme se snažit udržet počet vrstev na co nejmenší úrovni (maximálně 3-4 vrstvy), anebo zavést mezi vrstvami jistou synchronizaci (více kapitola 3.3).

Přístupová vrstva (FT) zprostředkovává (R) přístup ke zpětnému kanálu. Každý (FT) má ve své skupině přesně definovaný počet (R), aby všichni tyto (R) mohly odesílat RR pakety v nejkratším možném čase. Pro výpočet počtu (FT) v přístupové vrstvě zpětného kanálu slouží Rovnice 3.2.

$$FT_1 = \frac{N \cdot PL_1}{T_1 \cdot BW_{RTCP}} \quad [-] \quad (3.2)$$

kde

<b>FT<sub>1</sub></b>	[-]	definuje počet (FT) v přístupové vrstvě (vrstva 1)
<b>N</b>	[-]	celkový počet uživatelů zapojených do IPTV relace
<b>PL<sub>1</sub></b>	[bit]	velikost paketu RR, který je odesílán k FT1
<b>T<sub>1</sub></b>	[s]	definuje interval odesílání RR paketu (TRR)
<b>BW<sub>RTCP</sub></b>	[bit/s]	šířka pásma zpětného kanálu (3,75% celkové šířky pásma IPTV relace)

$$FT_2 = \frac{FT_1 \cdot PL_2}{T_2 \cdot BW_{RTCP}} = \frac{N \cdot PL_1 \cdot PL_2}{T_1 \cdot T_2 \cdot BW_{RTCP}^2} \quad [-] \quad (3.3)$$

Analogicky dostáváme pro vyšší vrstvy obecnou rovnici:

$$FT_H = \frac{N \cdot \prod_1^H PL}{BW_{RTCP}^H \cdot \prod_1^H T} \quad [-] \quad (3.4)$$

kde

<b>H</b>	[-]	značí výšku hierarchického stromu zpětné vazby
<b>FT<sub>H</sub></b>	[-]	definuje počet (FT) ve vrstvě H
<b>PL<sub>H</sub></b>	[bit]	velikost paketu RSI, který je odeslán k FTH
<b>T<sub>H</sub></b>	[s]	definuje interval odesílání RR paketu (TRR)
<b>BW<sub>RTCP</sub></b>	[bit/s]	šířka pásma zpětného kanálu (3,75% celkové šířky pásma IPTV relace)
<b>N</b>	[-]	celkový počet uživatelů zapojených do IPTV relace

Zpoždění se dá dále snižovat správnou synchronizací všech (FT), nastavení správného „časování“ odesílání RSI zpráv (více o tomto tématu naleznete v kapitole 3.3). Abychom dosáhli minimální doby průchodu paketu stromem, musí být čas pro odeslání RSI paketu ve všech vrstvách naprosto stejný. Nejprve uvedu princip minimalizace na příkladu 3vrstvého stromu a poté definici zobecním. S použitím Rovnice 3.4 a výšky stromu  $H = 3$ , dostáváme následující tvar rovnice:

$$FT_3 = \frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot T_1 \cdot T_2 \cdot T_3} \quad [-] \quad (3.5)$$

Po jednoduché úvaze, a to že v nejvyšší vrstvě stromu může existovat pouze samotný RFT ( $FT_H = 1$ ), dostáváme:

$$T_1 \cdot T_2 \cdot T_3 = \frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot FT_3} = \frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot 1} \quad [s] \quad (3.6)$$

Z Rovnice 3.6 pak vycházejí 2 základní předpoklady. Buď může definovat  $T_1$  podle standardu RTCP (5;  $\infty$ ), tudíž dostaneme Rovnice 3.7, která optimalizuje přenos pouze mezi vrstvami TTP stromu,

$$T_2 \cdot T_3 = \frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot T_1}$$

$$T_{opt}^2 = \frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot T_{RTCP}}$$

$$T_{opt} = \sqrt{\frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3 \cdot T_{RTCP}}} \quad [s] \quad (3.7)$$

anebo se od tohoto standardu odbourat a dostáváme maximálně optimalizovanou rovnici:

$$T_{opt} = \sqrt[3]{\frac{N \cdot PL_1 \cdot PL_2 \cdot PL_3}{BW_{RTCP}^3}} \quad [s] \quad (3.8)$$

Tato míra abstrakce se dá rozšířit na libovolně veliký strom a tak, bez dalšího vysvětlování uvádím přímo obecné rovnice. U Rovnice 3.9 je zachován vstupní požadavek RTCP intervalu pro odesílání RR paketu a Rovnice 3.10 představuje obecné vyjádření intervalu minimální doby pro odesílání RR paketu,

$$T_{opt} = \sqrt{(H-1) \frac{N \cdot \prod_1^H PL}{BW_{RTCP}^H \cdot T_{RTCP}}} \quad [s] \quad (3.9)$$

$$T_{opt} = \frac{\sqrt[H]{N \cdot \prod_1^H PL}}{BW_{RTCP}} \quad [s] \quad (3.10)$$

kde

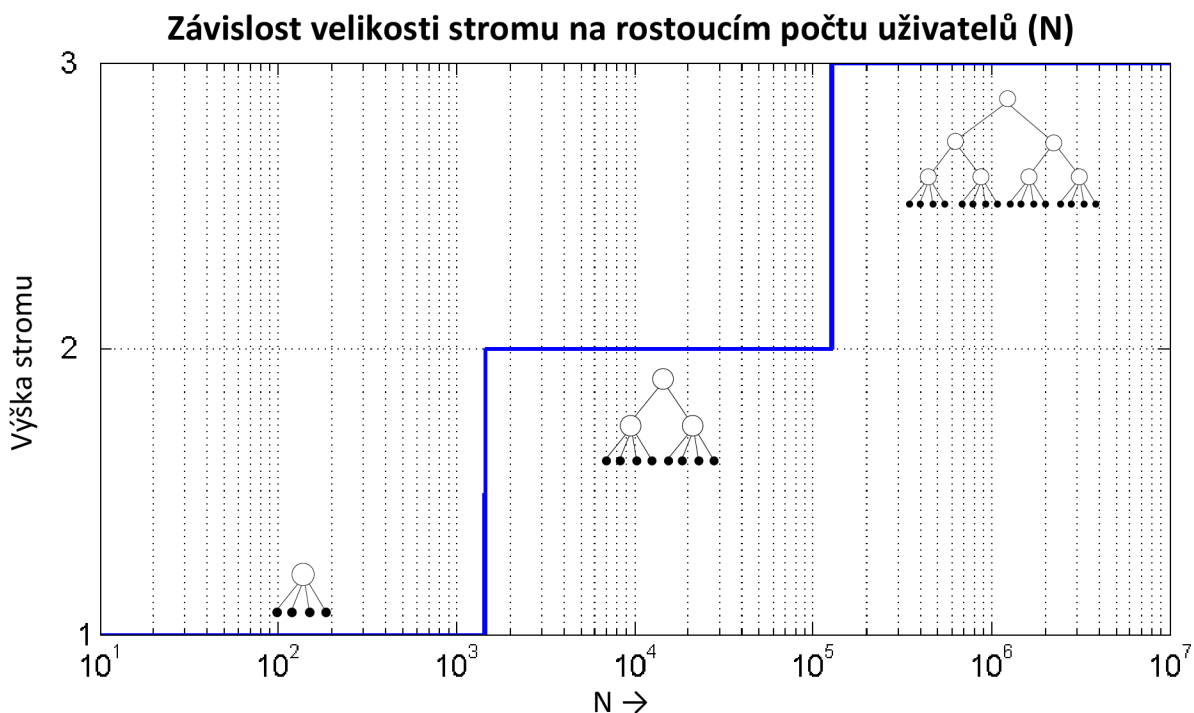
**H** [-] značí výšku hierarchického stromu zpětné vazby

## 3.2 Inicializace stromu

Proto, aby celá topologie stromu fungovala bez větších komplikací, musíme zajistit úvodní inicializaci stromu. Hlavní otázkou zůstává, kolik vrstev bude mít úvodní strom zpětné vazby, aby byl schopen pokrýt dostatečné množství uživatelů. Na tuto situaci jsem provedl simulaci pomocí programu Matlab<sup>1</sup> (Obr. 3.1), která zachycuje růst počtu vrstev hierarchického stromu v závislosti na přibývajícím počtu uživatelů. V grafu je vypočítávána výška stromu, která je potřebná pro daný počet uživatelů podle Rovnice 3.4. Rovnice vrací počet nutných (FT), aby byl zpětný kanál schopný pokrýt daný počet uživatel na základě požadavku relace. Výška stromu  $H$  se zvyšuje do té doby, než vypočtená hodnota (FT) bude menší než 1. Parametry, které byly použity za konstanty, jsou následující:

šířka pásma relace  $BW = 4\text{Mb/s}$ , celkové zpoždění přenosu RR zprávy v jedné vrstvě  $T_{\text{DELAY}} = 5\text{s}$ , velikost RR zprávy 480b a velikost RSI zprávy 8000b.

Velikosti RR a RSI jsou převzaty z [1] popřípadě [15]. Toto hodnoty již představují velice kvalitní přenos IPTV s dostatečnou odezvou i pro stávající RTP/RTCP protokol. Velikosti RSI paketu bude podrobněji věnována kapitola 3.3.4.



Obr. 3.1 - Úvodní inicializace vrstev hierarchického stromu

<sup>1</sup> <http://www.mathworks.com/>

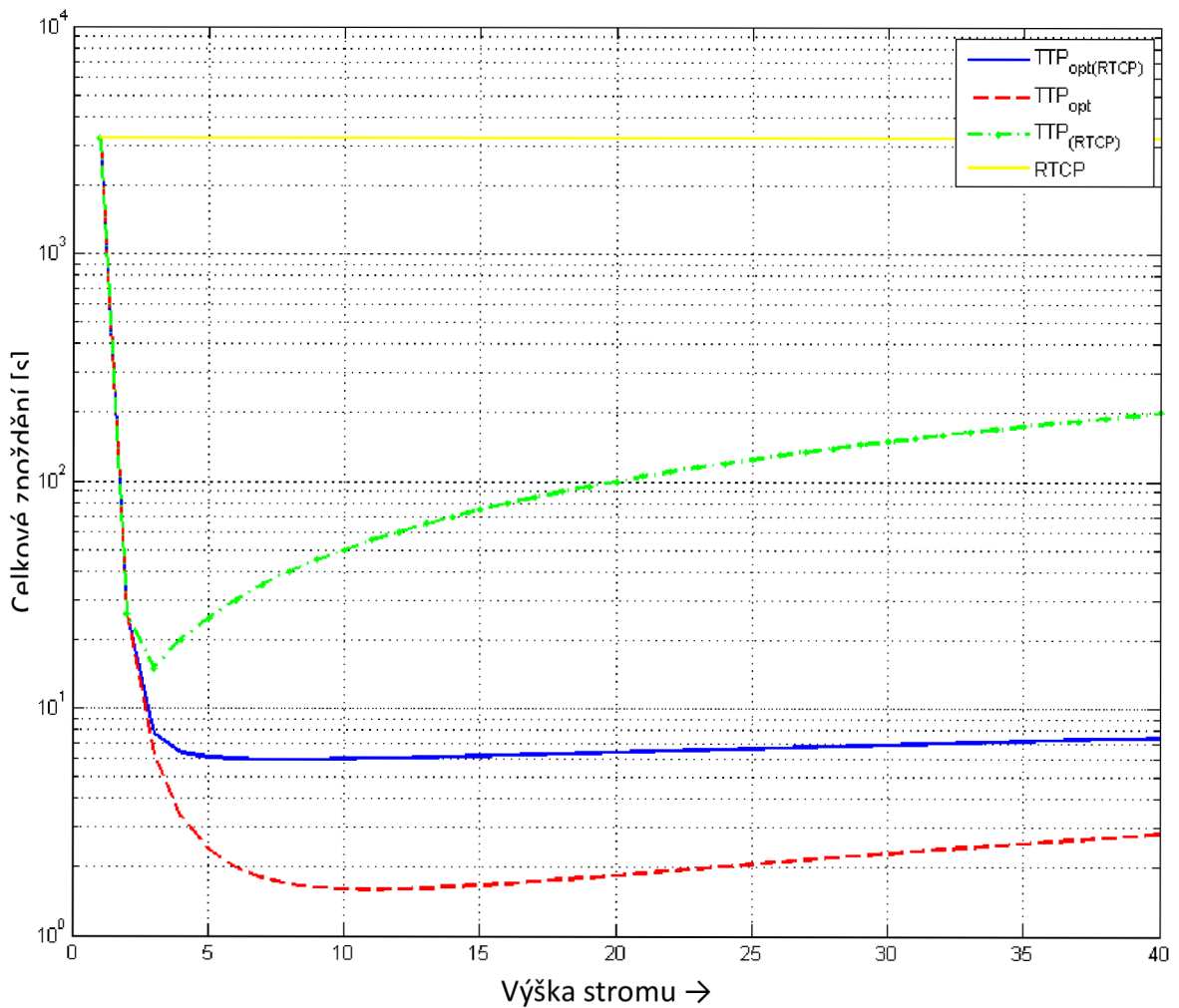
Ze simulace je patrné, že nemá cenu vytvářet strom, v kterém bude existovat pouze jedna vrstva (FT) (samotný RFT). Tato struktura je schopna obsloužit pouze malé množství uživatelů (asi 2000) jelikož představuje analogii je klasické RTP/RTCP relaci. Fungování samotného RFT ve stromu nepřináší žádný užitek do budoucna, kdy by došlo k rychlému připojení velkého množství členů do relace. Na druhou stranu je zase značně neekonomické vytvářet hned 3vrstvý strom, který je sice schopen pojmout obrovské množství uživatelů (více než  $10^7$ ), ale za cenu mnohem většího zpoždění (asi 15s) a hlavně celkově větší složitosti. Optimálním řešením pro úvodní inicializaci představuje 2vrstvý strom, který obsahuje v přístupové vrstvě potřebný počet (FT) a ti již odesílají sumarizovanou RSI zprávu k RFT. Celkové zpoždění udržíme ještě na rozumné hranici 10s a zpětný kanál je schopen za dané situace obsloužit až  $10^5$  členů.

### 3.3 Možnosti realizací

Na řadu přichází hned celá škála realizací takového přenosu paketů popsaného rovnicemi v kapitole 3.1. Základem TTP protokolu je H-vrstvý asynchronní strom, který bude detailně popsán v kapitole 3.3.1. Představuje jednoduchý model zpětného kanálu bez jakékoliv další synchronizace sestaveného stromu. Druhou možností představuje synchronizace zpětného kanálu všech vrstev (FT) (kapitola 3.3.2). Nevýhodou této metody je jednak vyšší složitost, ale mnohdy i větší počet (FT). Na druhou stranu dokáže sestavit optimalizovaný hierarchický strom, který je schopen urychlit čas průchodu RR zprávy celým stromem. Pro vyšší srozumitelnost bych rád na tomto místě uvedl simulaci provedenou v Matlabu, aby informace v dalších kapitolách dostaly reálnější podklad. Pomocí křivek vypočtených dle Rovnice 3.9 ( $TTP_{opt(RTCP)}$ ) a Rovnice 3.10 ( $TTP_{opt}$ ) znázorňuje Obr. 3.2 celkový čas potřebný pro přenos zprávy hierarchickým stromem v závislosti na jeho výšce. Počet vrstev stromu v Obr. 3.2 byl záměrně zvolen tak vysoký, aby bylo ukázáno, že existuje hranice počtu vrstev, po které celkový čas průchodu zprávy vytvořeným stromem již neklesá a dokonce začne narůstat. Parametry, které byly použity k simulaci, jsou následující:

$$BW = 4\text{Mb/s}, PL_1 = 480\text{b}, PL_{2,3,..} = 8000\text{b}, N = 10^7 \text{ a } T_{RTCP} = 5\text{s}.$$

## Celkové zpoždění zpráv v závislosti na výšce stromu



Obr. 3.2 – Optimalizace zpětného kanálu

Poznámka 1: na obrázku stojí za povšimnutí žlutá křivka, která definuje výsledné zpoždění v relaci pouze podle specifikace **RTP/RTCP**. Interval, po kterém (S) dostane zprávu od jednoho (R), kdy je v relaci připojeno 1 000 000 klientů, přesahuje hranici **3000 sekund**. V porovnání s protokolem TTP je tato hodnota více než **300x horší**. Při použití plně optimalizovaného zpětného kanálu o 10 vrstvách (FT) je tato hodnota dokonce až **3000x horší**. Ovšem za cenu několikanásobně většího počtu (FT)

Poznámka 2: Velikost sumarizačního paketu RSI je stanovena na základě výzkumu [15]. O velikosti RSI paketu více pojednává kapitola 0, pro tuto simulaci ji, prosím, berte jako hodnotu definující velice podrobné vlastnosti statistiky.

### 3.3.1 Asynchronní zpětný kanál

Nyní se pojdme v této kapitole zaměřit především na asynchronní přenos zpráv zpětným kanálem, který je v grafu na Obr. 3.2 vyneseno pomocí zelené a červené křivky. Asynchronní přenos má tu výhodu, že zpětný kanál není třeba vůbec synchronizovat k tomu, aby poskytoval optimální službu. Každý (FT) přijímá stejná data o stavu stromu jako (R) z multicastu (kanálu SR) a navíc je informován zprávami FTD od (FTM) o tom, jak velkou skupinu (R) podporuje. Z těchto dat je každý člen schopen vypočítat odpovídající čas pro to, kdy odeslat sumarizovanou zprávu RSI. Asynchronní kanál vznikne tehdy, pokud zvolíme stejný časový interval ve všech vrstvách včetně přístupové, ke které se připojují (R). (FTM) je ten prvek v síti, který rozhodne podle zadaných parametrů, jak vytvoří zpětný kanál. Ostatní členové zpětného kanálu již jen dodržují dohodnutá pravidla.

Stejný časový interval definovaný podle RTCP ve všech vrstvách hierarchického stromu je značně limitující, ale na druhou stranu přináší i svá pozitiva. Limity jsou dány především v délce doby přenosu RR zprávy. S výškou stromu totiž lineárně roste zpoždění, které zpětný kanál do komunikace vnese.

#### 3.3.1.1 S dodržováním parametrů RTCP

Jak již bylo řečeno, pokud bude hierarchický strom dodržovat podmínku RTCP intervalu, bude sice jeho čas průchodu stromem růst rychleji, ale za to nebude třeba tolika sumarizačních uzlů (FT). Nejlépe to dokazuje Obr. 3.3. Sami vidíte, že počet vrstev se pro jeden milion klientů zastavil již třech. Tato skutečnost se projeví optimálně právě v celkovém počtu (FT) potřebných pro realizaci celého stromu. Počet (FT) je na Obr. 3.3 naznačen zelenými křížky.

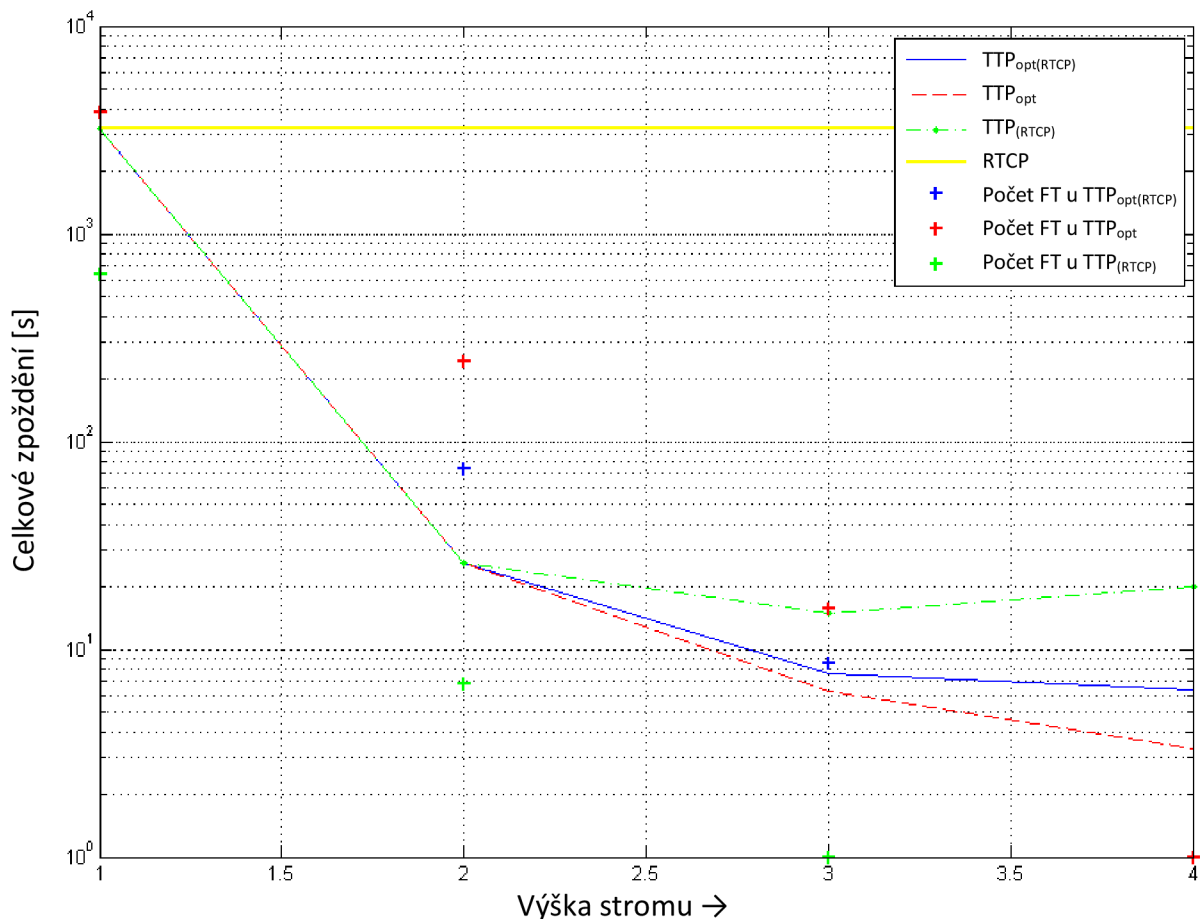
#### 3.3.1.2 Bez dodržování parametrů RTCP

Pokud nám však záleží na rychlosti přenosu zprávy ve zpětném kanále a máme v úmyslu použít asynchronní kanál, nezbude nám jiné východisko, než uměle zkrátit interval odeslání odesílání. Termínem „uměle zkrátit“ je myšleno aktivovat větší počet (FT) v přístupové vrstvě hierarchického stromu, aby se ke každému (FT) mohl přihlásit menší počet (R) a ty tak mohly svoji zprávu posílat častěji. Interval odesílání RR paketu pak již neodpovídá RTCP specifikaci a má rozměr  $(0; \infty)$ . Tato metoda s sebou však nese spoustu nesnází. První vlastností je obrovský počet (FT) pro realizaci IPTV pro velkou skupinu uživatel. Na Obr. 3.3 je zřejmé, že k tomu abychom 1 000 000 klientům poskytli rychlý interval doručení zprávy, potřebujeme obrovský počet (FT), který není zřejmě žádný poskytovatel schopen realizovat. Odezva necelé 3s podle Obr. 3.3 je sice naprosto nesmyslná hodnota k provozování takové služby, ale alespoň dokládá enormní počet (FT), kterých je třeba pro realizaci 4vrstvého hierarchického stromu. Další problém nastává

s množstvím generovaných dat, které je v síti třeba zpracovat. I když je na to síť připravena co do počtu členů, je třeba pamatovat na jejich vyšší výpočetní výkon, jelikož budou muset být schopni toto množství zvládat, aby celkový chod služby byl dostatečně kvalitní.

Samozřejmě však mohou existovat situace, kdy se takového stavu sítě využije a to především v situacích, kdy v relaci nebude připojeno příliš velké množství uživatelů a byla by škoda dostupné zdroje nevyužít pro zvýšení kvality celé služby. Nesmíme také zapomínat na skutečnost, že protokol TTP není bezprostředně spjat se službou IPTV. Může být stejně dobře využit v rámci úplně jiné služby, u které využijeme jeho schopnosti sběru dat v reálném čase s mnohem větší dostupnou šířkou pásma pro zpětný kanál. U IPTV totiž z celkové šířky pásma máme k dispozici pouze 3,75%.

### Celkové zpoždění závislé na výšce stromu

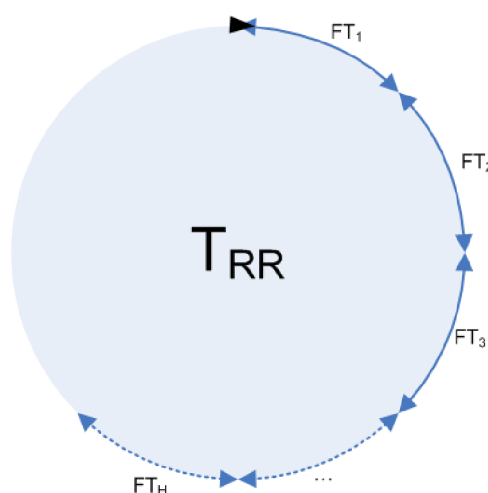


Obr. 3.3 - Počet (FT) pro synchronní a asynchronní zpětný kanál



### 3.3.2 Synchronní zpětný kanál

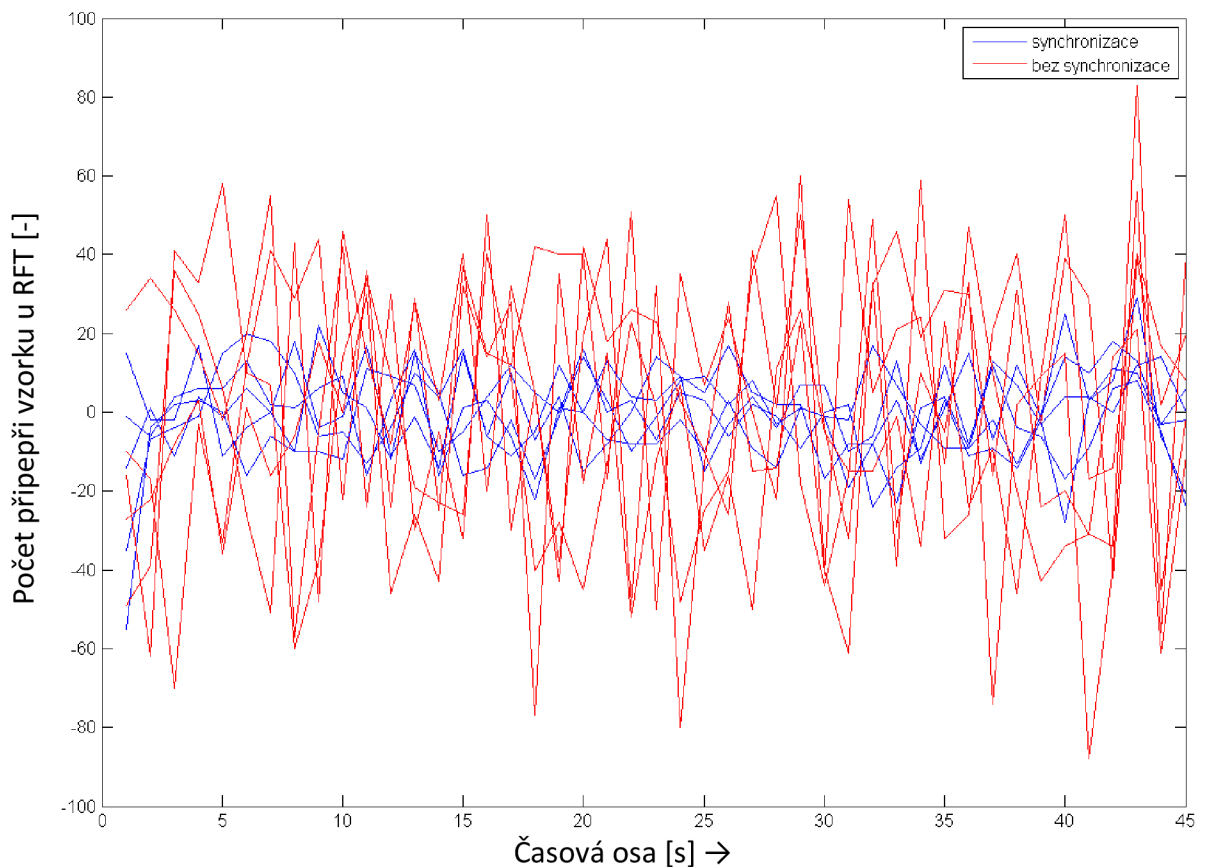
Synchronní zpětný kanál představuje alternativu k přenosu asynchronnímu. Je schopen za určitých podmínek vylepšit dobu přenosu zprávy ve zpětném kanále. Jedná se zejména o případy, kdy máme k dispozici nevyužitou výpočetní kapacitu (nečinné FT) a přitom relace požaduje zachování parametry RTCP protokolu u přístupové vrstvy. K tomu je však potřebná vzájemná synchronizace mezi (FT) a (FTM), kdy všichni aktivní (FT) musí mít správně definovanou časovou osu, na které volí svůj čas odesílání RR zpráv. Doslova se dá říci, že si aktivní (FT) nesmí zvolit začátek intervalu samovolně, ale musí být volen dle časového harmonogramu tak, aby na sebe jednotlivé intervaly v sousedních vrstvách navazovaly (Obr. 3.4). Tím je dosaženo maximální efektivity při přenosu zpráv ve zpětném kanále. Výsledek simulace takového provedení představuje modrá křivka na Obr. 3.3 a Obr. 3.2. Na nich je definován interval pro odesílání RR paketu podle standardu RTCP, ale zbytek přenosu ve stromu je zcela ponechám k optimalizaci (FTM). Matematickou realizaci představuje Rovnice 3.9 a především z Obr. 3.2 je vidět, že takovýto strom dokáže i při velkém množství uživatelů celkový čas přenosu zpětný kanálem jen nepatrně prodloužit (v závislosti na počtu vrstev hierarchického stromu). I zde však dochází k použití většího počtu (FT). Tento počet ale není tak velký jako v případě optimalizovaného asynchronního stromu.



Obr. 3.4 - Posunutí intervalu generování a odeslání RSI zpráv u vyšších vrstev hierarchického stromu

Další nespornou výhodou představuje nižší rozptyl počtu RSI zpráv, které jsou v každém intervalu získány od nižší vrstvy (FT) ku počtu očekávaných zpráv. Ke každému (FT) by se v definovaném časovém intervalu mělo v průměru dostat stejné množství zpráv. Zprávy jsou však generovány s náhodným násobkem z intervalu  $\langle 0,5;1,5 \rangle$  dle [1], tudíž je jejich příchod v krátkém čase nepředvídatelný. Kolísání počtu RSI zpráv pro synchronní a asynchronní strom je zobrazeno na Obr. 3.5.

### Rozdíl počtu přijatých RSI zpráv ku počtu zpráv očekávaných

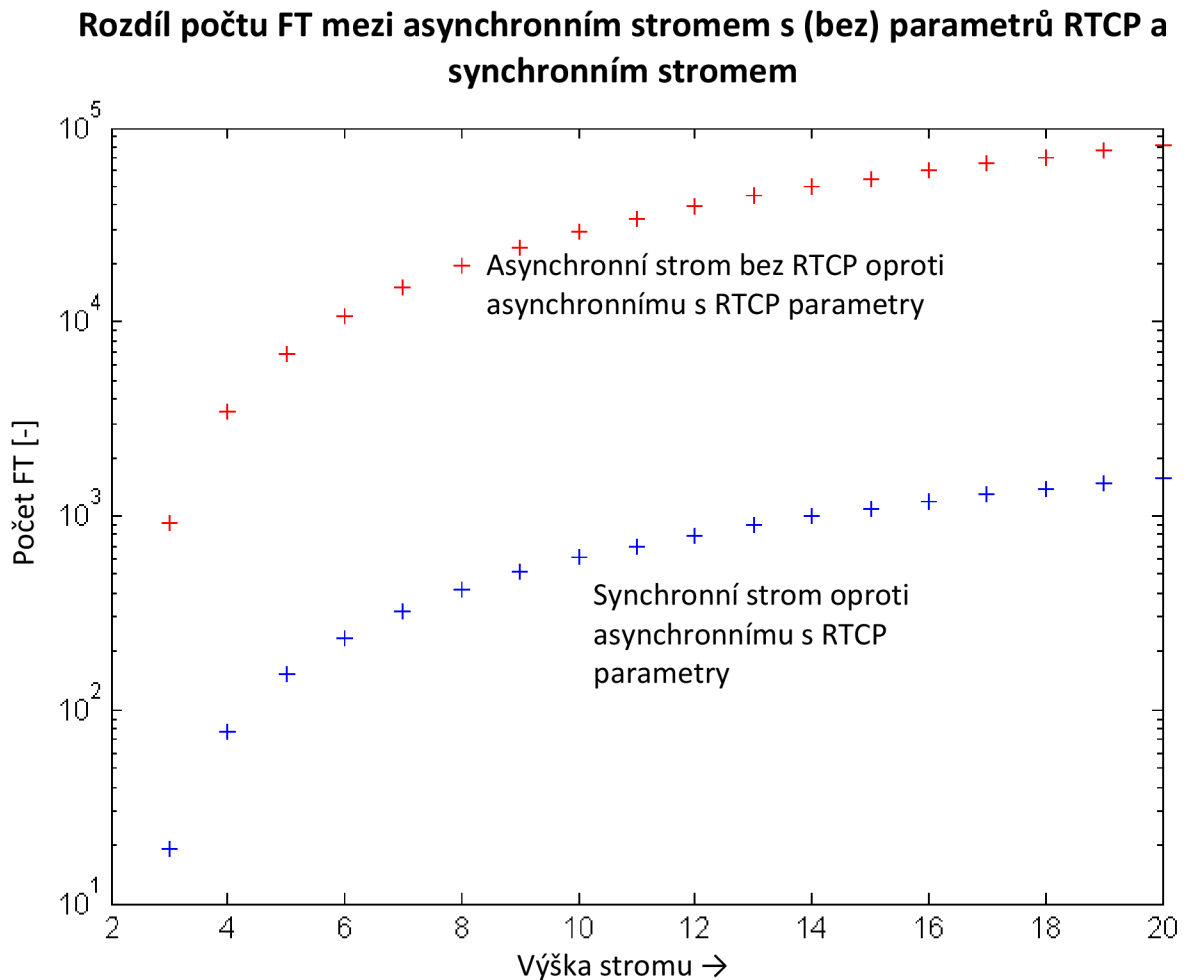


Obr. 3.5 - Rozptyl počtu příchozích RSI zpráv

### 3.3.3 Využití více vrstev hierarchického stromu

Pokud máme pro realizaci dostatečný počet (FT), můžeme si dovolit vytvářet H-vrstvé hierarchické stromy téměř bez omezení. Jak bylo ukázáno na Obr. 3.2, počet vrstev promlouvá výrazně do celkového zpoždění přenosu zprávy zpětným kanálem. Pokud je množství informace, které hodláme přenést, dostatečně velké, zpočátku přidáním další vrstvy (FT) podle Rovnice 3.4 dosáhneme snížení intervalu pro posílání sumarizovaných zpráv mezi vrstvami. Vždycky však existuje hranice, po které přidání další vrstvy zlepšení nepřináší, anebo je tak minimální, že nepřináší v danou chvíli užitek. Nejlépe je to asi vidět opět na Obr. 3.2, kde pro červenou i modrou křivku je minimum funkce natolik ploché, že můžeme stanovit téměř stejné zpoždění přenosu paketu pro hierarchické stromy s počtem vrstev od 5 do 20. Další nepříjemnost přináší samotný počet (FT), kterých je zapotřebí pro danou optimalizaci. Pro tento případ jsem opět připravil simulaci Obr. 3.6, kde je vidět rozdíl v počtu (FT), kterých je zapotřebí při realizaci synchronního kanálu s dodržováním parametru RTCP nebo bez jeho dodržování. Simulace je provedena pro relaci s 1 000 000 uživatelů, pro které se asynchronní zpětný strom skládá ze 3 vrstev a potřebuje celkem 640 (FT) v přístupové vrstvě, 7 (FT) v nadřazené

vrstvě a 1 (RFT). Všechny parametry IPTV relace zůstávají stejné jako na Obr. 3.2 a Obr. 3.3. Jak je vidět z modrého průběhu na Obr. 3.6, již na 3 vrstvách dokáže synchronní strom s přidáním pouhých 20 (FT) poskytnout rychlejší přenos zprávy od (R k (S). Toto vylepšení představuje zkrácení času průchodu z 15 sekund u asynchronního stromu na 8 sekund (Obr. 3.3) u synchronního stromu s dodržáním parametrů RTCP na přístupové vrstvě.



Obr. 3.6 - Rozdíl počtu sumarizačních uzlů (FT)

### 3.3.4 Definování velikosti sumarizačního paketu

Velikost sumarizačního paketu (RSI paketu) je vždy definována aktuální stavem zpětného stromu, požadavkem na detailnost získané informace o stavu příjmu a počtem uživatelů, kteří jsou v relaci zapojeni. RSI paket je náhražkou původního RR paketu, kde se v jednom paketu přenáší informace od více uživatelů. V rámci RSI paketu mohou být zprávy ve zpětném kanále přenášeny mnohem efektivněji. Není třeba vybavovat každou informaci vlastní hlavičkou, ale je lepší zprávy sloučit dle definované šablony a odeslat je. Již v úvodní kapitole 2.1.2 byla definována velikost RSI paketu dle Rovnice 2.1. Na tomto místě je vhodné definici ještě více upřesnit. Podle [15] a [2] je velikost hlavičky RSI rovna 352bitům. Další bloky představují potřebnou velikost k uložení informace od všech uživatelů do intervalu vlastností, které RSI pošle dále. Každý interval musí být připravený na maximální počet uživatelů  $N$  v relaci. Vlastnost má opět pevný počet 96bitů a obsahuje informaci o tom, jaký druh informace a v jak jemném intervalu má být přenesen. Dosazením do Rovnice 2.1 dostaneme následující rovnici:

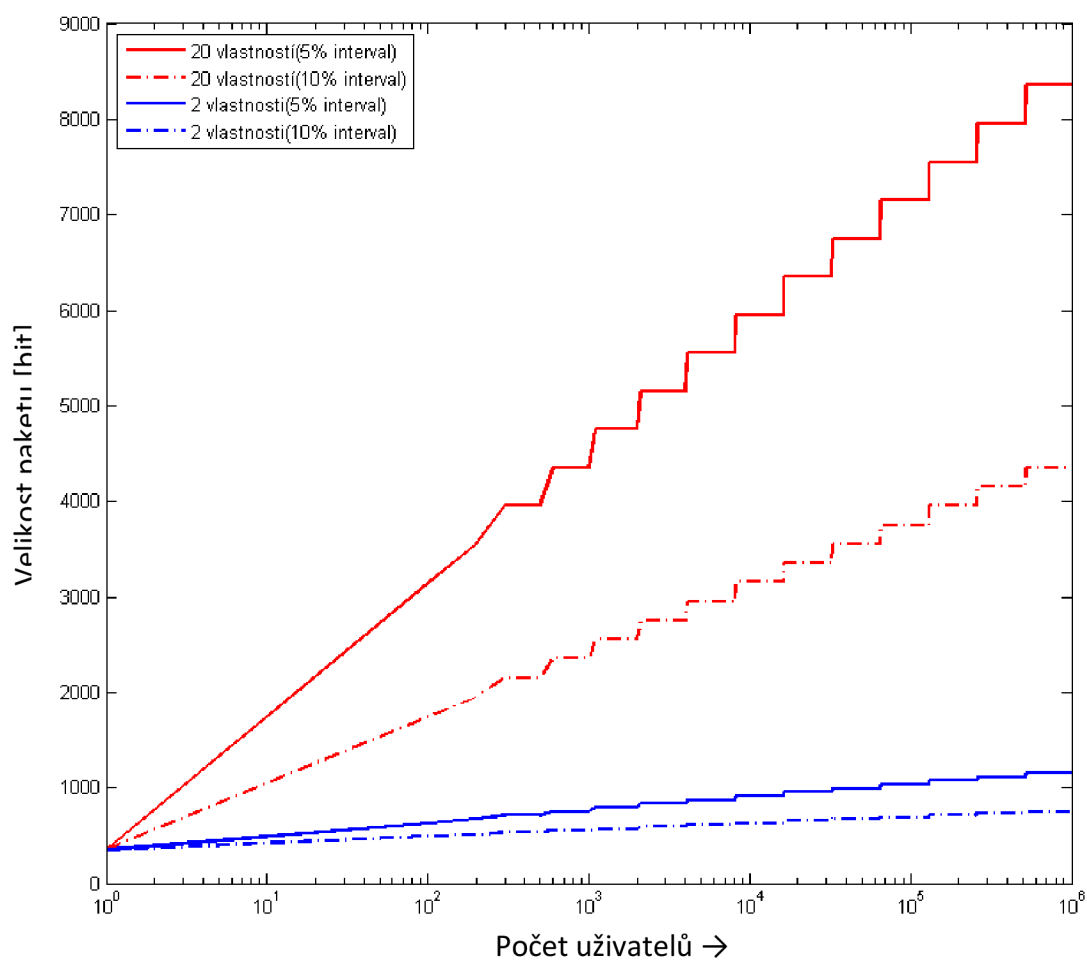
$$PL_{RSI} = 352 + \log_2(N) \cdot \left\{ \sum (96 \cdot \text{Počet\_intervalů}) \right\} \quad [s] \quad (3.11)$$

kde

$PL_{RSI}$	[bit]	značí velikost RSI paketu
$N$	[-]	počet uživatel
$\text{Počet\_intervalů}$	[-]	definuje jemnost intervalu, na který je dělena určitá vlastnost

Simulace provedená opět v prostředí Matlabu ukazuje na Obr. 3.7 růst velikosti sumarizačního RSI paketu v závislosti na počtu uživatelů a počtu vlastností. Z Rovnice 3.11 je zřejmé, že závislost je logaritmická a závisí na počtu uživatelů a množství dat, který chceme přenášet.

## Závislost velikosti RSI paketu na počtu uživatelů



Obr. 3.7 - Velikost RSI paketu

Jak je vidět z grafu na Obr. 3.7 velikost sumarizačního paket (RSI) je značně závislá na počtu uživatelů. Čím méně jich je, tím méně potřebujeme bitů, ke tvorbě správného RSI paketu. S řízenou velikostí paketu lze opět dosáhnout další optimalizace. Není třeba ve všech vrstvách stromu vytvářet stejně velký sumarizační paket. Každá (FT) zná velikost své supiny a může tedy naprosto přesně zmenšit danou velikost paketu.

## 4 PRAKTICKÁ REALIZACE ZPĚTNÉHO KANÁLU

Protokol TTP se od původního návrhu a specifikace dosti změnil, ale základní myšlenky samozřejmě zůstaly zachovány. Jedná se především o sběr RR paketů v sumarizačních uzlech a následně jejich distribuce ve stromu pomocí hierarchické agregace. TTP byl původně zaměřen pouze na úzkou skupinu služeb jako je IPTV. Postupem času se však ukázalo, že je ho možné použít pro všeobecný sběr dat, a tak se stal samostatným aplikačním protokolem, který lze využít pro celou řadu situací. Příkladem lze uvést sběr dat z měřících stanic, sběr dat o kvalitě vody v určitém kritickém úseku, internetová radia, internetová videotelefonie, multimediální multiplayerové hry a samozřejmě IPTV, která se stala hlavní motivací pro vznik TTP protokolu.

Původní návrh doznal velkých změn především ve způsobu sestavení zpětného kanálu. Z původně navržených paketů zůstali zatím pouze 3, která obstarávají činnost celého protokolu. Byla změněna část parametrů a byly přidány nové vlastnosti, jako je možnost časové synchronizace hierarchického stromu, která není ovšem zatím uspokojivě vyřešena. Již od začátku je celý vývoj protokolu veden v programovacím jazyku Java [20] a podílí se na něm několik lidí. Zodpovědnost za samotnou realizaci byla svěřena mně a mému vedoucímu práce. Oba dva pracujeme na rozdílných částech protokolu, ale provádíme velice časté konzultace o směru vývoje. Mým úkolem se stalo, jak je patrné z této práce, sestavení optimálního zpětného kanálu a realizace metody, která se bude starat o jeho sestavení, správu a následné řešení kolizních stavů. Jedním z účelů dosavadního textu, bylo poskytnout čtenáři dostatečnou představu o komplexnosti a také komplikovanosti celého řešení. Rád bych na tomto místě podotkl fakt, že až doposud se jedná zcela výhradně o mé myšlenky a nápady, kterými jsem po konzultaci s Radimem Burgetem a Danem Komosným rozšířil a obohatil návrh TTP protokolu. Zpočátku bylo hodně počítáno s možností využití sítě PlanetLabu [21], které jsme členem v rámci sítě CESNET, a kde máme možnost využívat téměř všech jeho uzlů. V současné době se jejich počet pohybuje kolem 850 uzlů<sup>2</sup>. Ukázalo se však, že na vývoj tohoto druhu protokolu není síť PlanetLabu zpočátku příliš vhodná, jelikož představuje pro testovací účely příliš složitou organizaci a správu. Proto byla naším týmem vytvořena simulační knihovna JSimLib3, který nám toto omezení pomáhá překlenout.

---

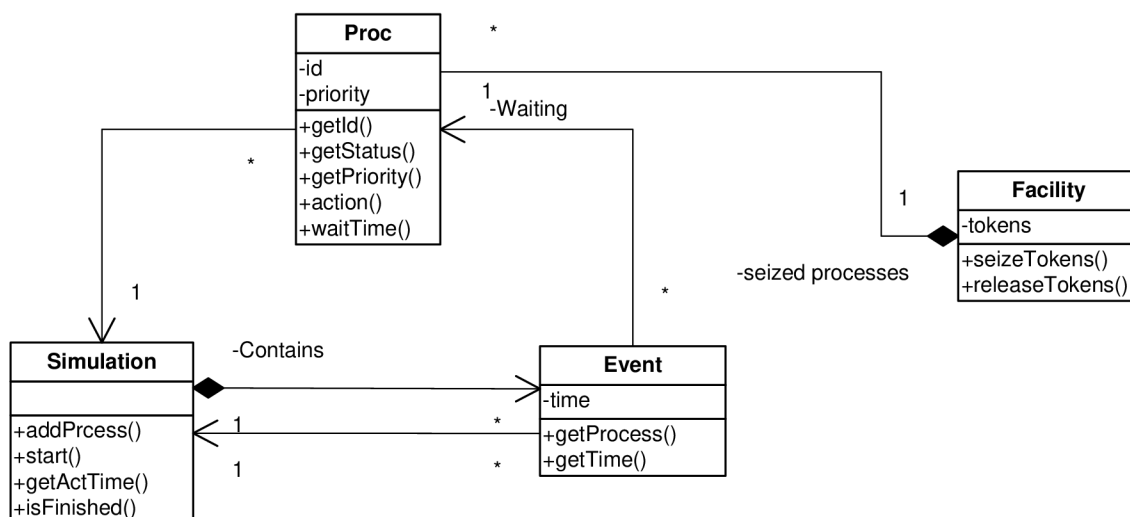
<sup>2</sup> Informace z 9.5.2008 [www.planet-lab.org](http://www.planet-lab.org)

## 4.1 Simulační knihovna JSimLib3

Knihovna JSimLib3 pro náš vývoj představuje prostředí pro simulaci a zároveň testování různých prvků, které chceme v reálné síti využít. Snaží se definovat jejich chování, tak jako v reálné síti. Lze v ní specifikovat objekty, jejichž procesy jsou knihovnou synchronizovány a řízeny tak, že téměř nepoznáme, zda pracujeme s reálnou sítí nebo se simulací. Každý objekt vlastní minimálně jeden proces, který vykonává určitou akci. Objekt lze v simulaci spustit s libovolným časovým zpožděním, které je definováno při jeho inicializaci. Aplikace je navržena jako multivláknová a provádění jednotlivých procesů v diskrétním čase je paralelní, popřípadě pseudo-paralelní, v případě že procesů je více než kolik architektura, na které je simulace nasazena, umožňuje.

### 4.1.1 Základní struktura knihovny

Základní bloková struktura knihovny JSimlib3 je vyobrazena v notaci jazyka UML [16] na Obr. 4.1. Jedná se o celkem 4 základní stavební bloky: simulaci (třída „Simulation“), proces (třída „Proc“), událost („Event“) a zařízení (třída „Facility“). Simulace udržuje informaci o aktuálním čase (getActTime) a udržuje množinu naplánovaných událostí. Časovač je během simulace inkrementován vždy o jednotku času nahoru, a jestliže existuje nějaká událost naplánovaná na aktuální čas, je provedeno spuštění procesů, ke kterým události patří.



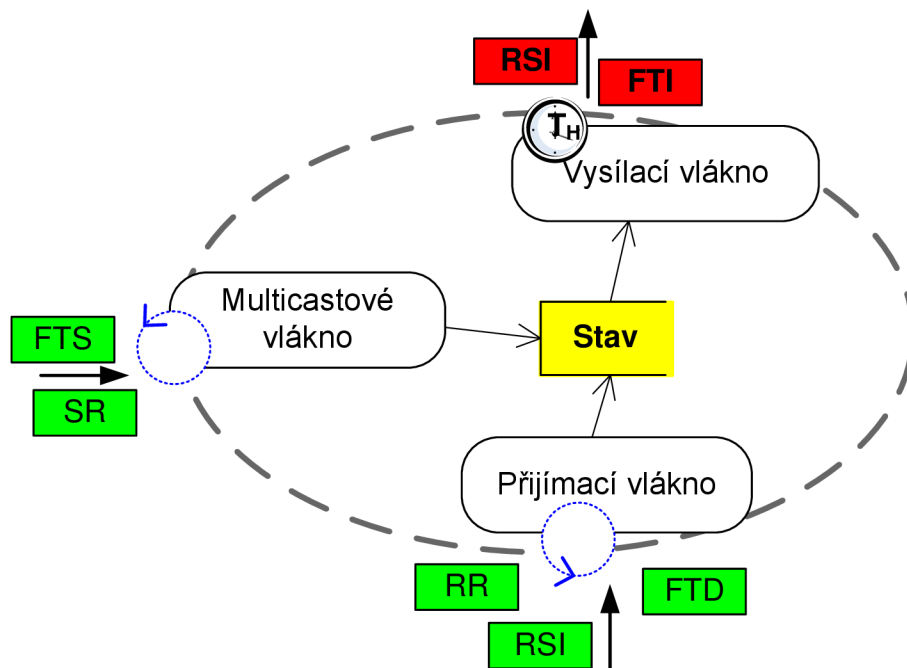
Obr. 4.1 - Struktura knihovny v UML notaci

Důvod vzniku simulační knihovny je prozaický. Knihovna byla navržena tak, aby co nejdříve napodobovala chování reálné sítě. Pro náš projekt pak platí, že ve chvíli kdy objektům přiřadíme místo třídy „Proc“, která je řídicím procesem všech objektů, třídu „Thread“, nezačnou se naše pakety posílat jen v rámci simulace, ale budou odeslány

rovnou do reálné sítě. Odladění případných chyb poté zabere mnohem méně času, než v případě kdybychom využili jiných simulačních prostředí. Právě z tohoto důvodu jsme pro začátek opustili myšlenku provádět všechny simulace v prostředí sítě PlanetLab, protože bychom tím náš návrh zbytečně zkomplikovali.

#### 4.1.2 Komunikace mezi procesy

Další velice důležitou věcí je vzájemná komunikace mezi procesy jednoho objektu. Na Obr. 4.2 je zakreslena činnost (FT) při příjmu a odesílání zpráv. Každý (FT) obsahuje 3 vlákna (přijímací, vysílací, multicastové) a úložiště dat, které je společné pro všechny 3 procesy. Na Obr. 4.2 je toto úložiště pojmenováno „Stav“. Přijímací a Multicastové vlákno pracují neustále a jakmile se na jejich rozhraní objeví paket určený pro ně, provedou patřičnou akci. Tímto podmětem může být příjem RR zprávy nebo RSI zprávy anebo příjem paketů FTD,FTI a FTS, které spravují celý zpětný strom. Podrobně budou jednotlivé akce popsány v kapitole 4.3. Vysílací vlákno odesílá data v předem definovaném časovém intervalu podle Rovnice 3.9 nebo Rovnice 3.10, který po čase  $T_H$  odešle RSI zprávu. V případě nutnosti odeslání FTI zprávy smí (FT) toto zprávu odeslat neprodleně.



Obr. 4.2 - Komunikující procesy v rámci sumarizačního uzlu (FT)



## 4.2 Sekvenční diagramy

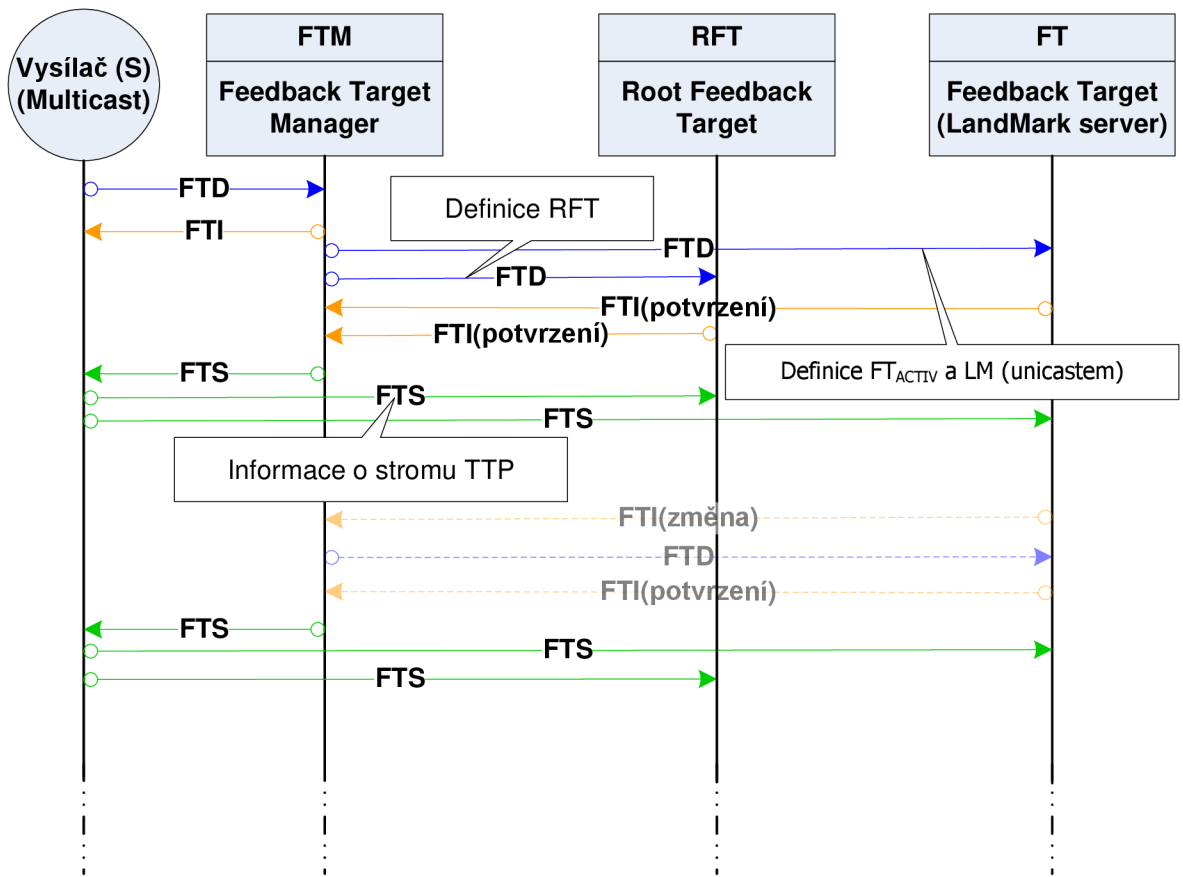
V této části budou uvedeny sekvenční diagramy jednotlivých prvků sítě v notaci jazyka UML [16]. Popisování čistého kódu by bylo totiž značně nepřehledné a matoucí. Na Obr. 4.3 a Obr. 4.5 je popsán časový sled událostí, které odpovídají přeposílání zpráv mezi jednotlivými prvky zpětného kanálu. Kapitola 4.2.1 pojednává o procesu sestavení a správy hierarchického stromu zpětné vazby a kapitola 4.2.2 o provozu v sestaveném zpětném kanále. Obě kapitoly jsou doplněny o Obr. 4.4 a Obr. 4.6, které dále popisují sekvenční diagramy z hlediska komunikace mezi členy. Z důvodu přehlednosti byly sekvenční diagramy zkráceny a jejich kompletní znění lze nalézt v dodatku na konci práce nebo na přiloženém CD.

### 4.2.1 Sestavení zpětného kanálu

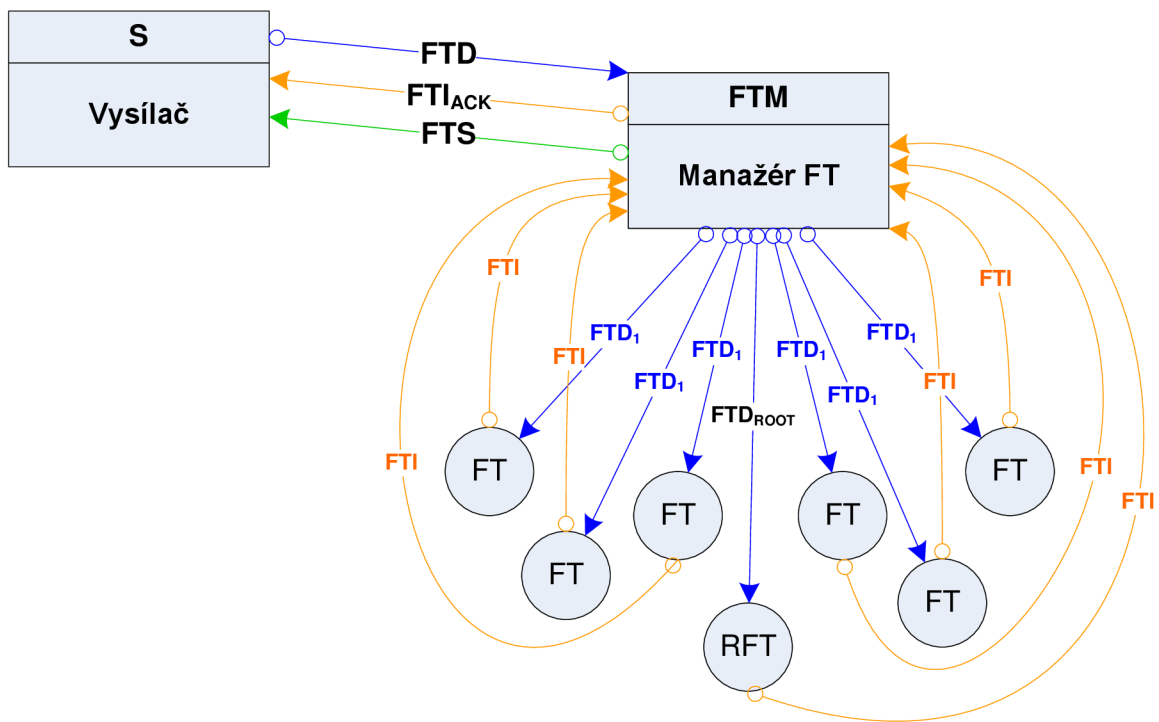
Počáteční sestavení hierarchického stromu hraje zásadní roli pro následnou funkčnost celého protokolu. Tato problematika byla popsána především v kapitole 3.1 a proto se nyní podíváme rovnou na realizaci takového stromu. Proces je rozdělen do několika fází, kde hned v první (FTM) získá od (S) informace o druhu služby a základních parametrech (především BW). Pro větší názornost byl připraven časový diagram, který lze nalézt na Obr. 4.3.

(FTM) má dokonalý přehled o tom jaké servery má k dispozici a o tom jaká je jejich aktuální činnost. Než však zjistí, kde a jaký počet klientů bude využívat službu poskytovanou (S), provede úvodní realizaci 2-vrstvého stromu. Toto provede odeslání zprávy FTD na adresu vybraných (FT), který se stanou aktivním sumarizačním uzlem. Tento proces je popsán na Obr. 4.11. Dále pak definuje RFT, který bude shromažďovat sumarizované zprávy od aktivních (FT) a předávat je (S) k vyhodnocení. Vybraní (FT), kteří byli požádáni o přechod do stavu aktivního (FT), odešlou potvrzující zprávu FTI a nastaví si všechny definované parametry obdržené v paketu FTD. To samé provede (RFT). Tím je strom prakticky již realizován a je schopen přijímat zprávy a vytvářet zprávy sumarizované (RSI).

IP adresu nadřazeného (FT) si musí již každý (FT) získat sám a to z SR kanálu. (FTM) sem totiž vysílá kompletní stav o relaci ve formě FTS paketů. Sekvenční diagram sestavení zpětného kanálu naleznete na Obr. 4.3. Je zde zachycena i situace, kdy jeden nebo i více (FT) zjistí překročení definovaného stavu uživatelů, na které byl definován a pošle tuto informaci (FTM), aby vypočítal nový stav stromu. Na Obr. 4.4 je pak sekvenční diagram rozkreslen do stavového. Je zde podrobněji ukázána komunikace mezi jednotlivými členy TTP protokolu, kdy je tvořen 2vrstvý hierarchický strom s jedním (RFT) a 6 (FT) v přístupové vrstvě.



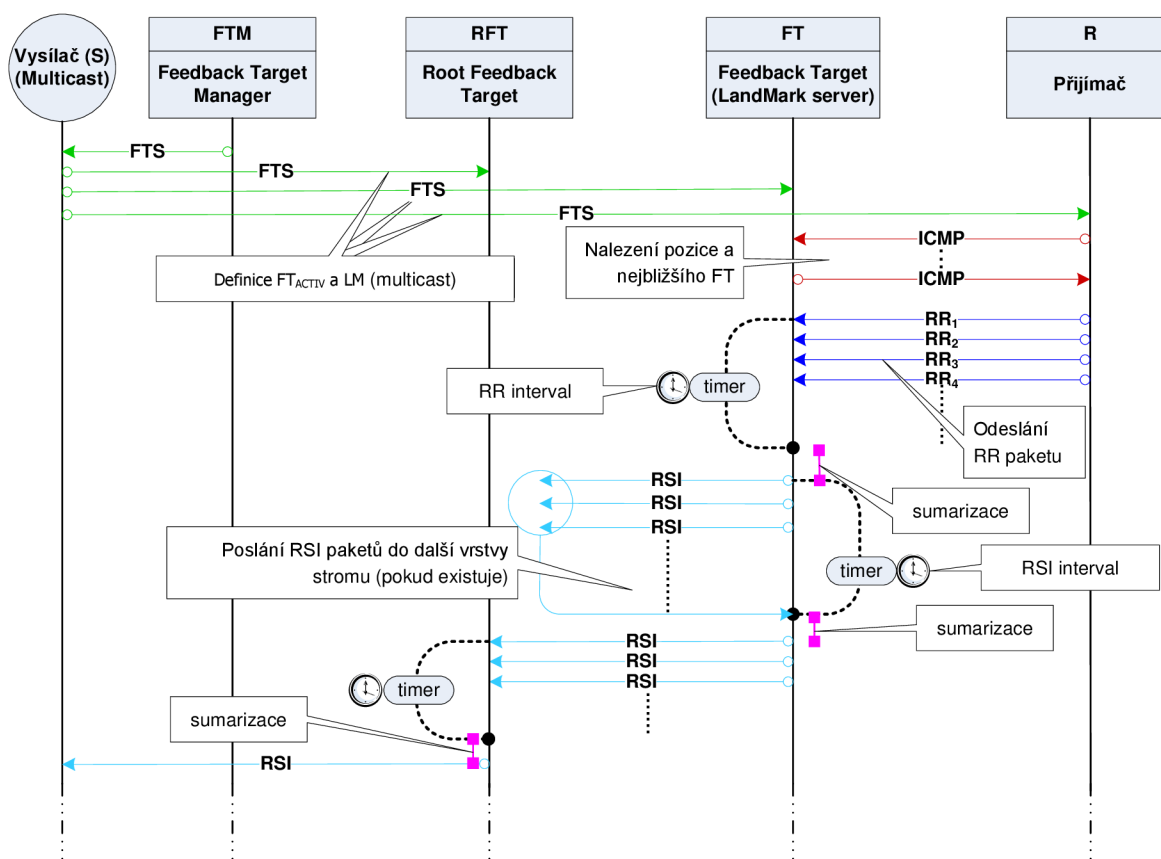
Obr. 4.3 - Sestavení hierarchického stromu zpětného kanálu



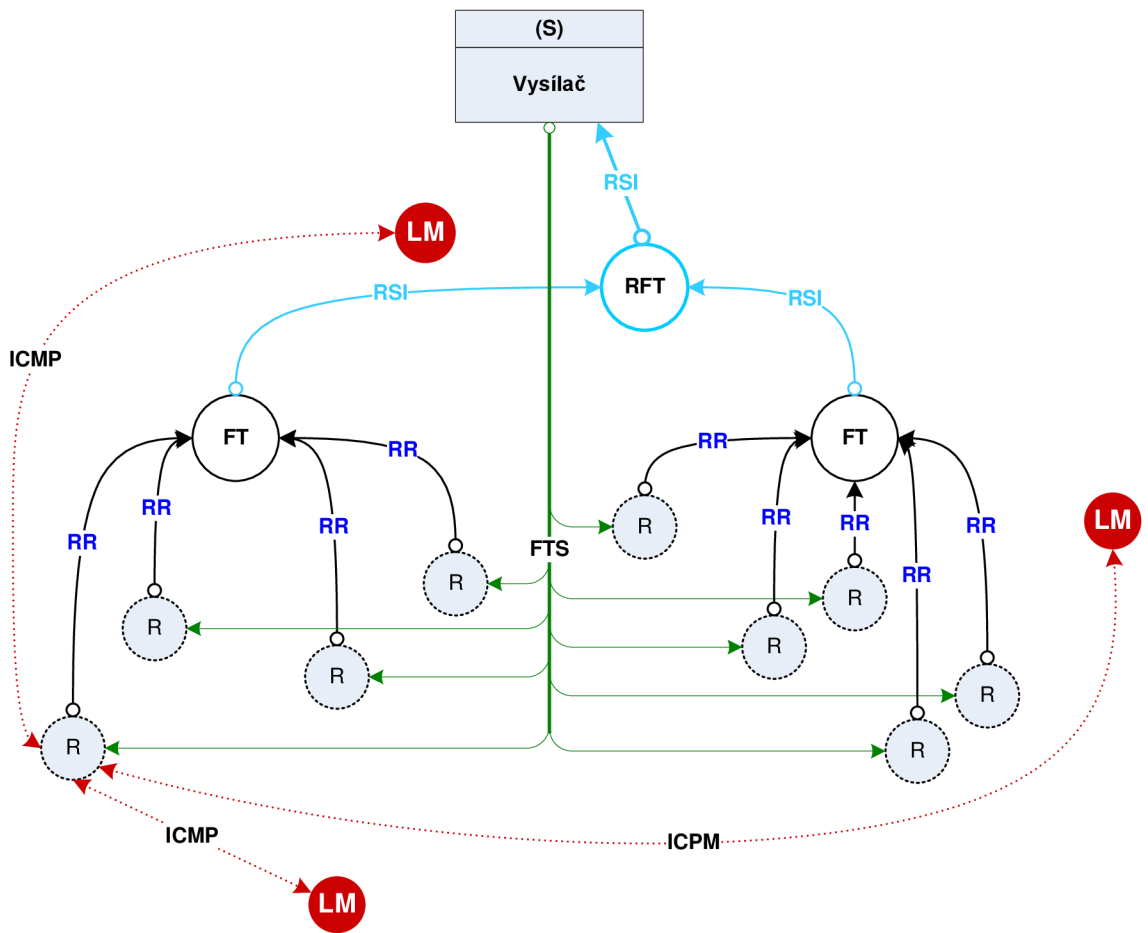
Obr. 4.4 - Komunikace mezi prvky zpětného kanálu při vytváření hierarchického stromu

## 4.2.2 Využití zpětného kanálu uživateli

Do vytvořeného kanálu poté začnou uživatelé vysílat své zprávy o kvalitě služby. K tomu, aby věděli k jakému sumarizačnímu uzlu paket odesílat, musejí nejprve najít svoji pozici a určit se sady aktivovaných (FT) toho nejbližšího, jelikož ten byl s největší pravděpodobností vybrán (FTM) jako jejich přidružený cíl zpětné vazby. (R) neustále sledují SR kanál a v případě jakékoliv změny provedou nové hledání nejbližšího (FT). Na Obr. 4.5 je naznačen sekvenční diagram takového procesu. (FTM) nejprve vytvoří síť a potvrdí si od všech (FT), že jsou schopni vykonávat požadovaný stav definovaný v FTD paketu. Poté odešle do SR kanálu informaci o stavu všech (FT), kteří tvoří zpětný kanál. Podle této informace zjistí každý (R) k němu nejbližší (FT) a začne odesílat v definovaném intervalu  $T_{opt}$  (Rovnice 3.9 nebo Rovnice 3.10) vytvořené RR zprávy. Každý (FT) ve stejném intervalu zprávy RR přijímá a tvoří z nich RSI zprávu, kterou následně odešle do další vrstvy hierarchického stromu, až zpráva nakonec doputuje k (RTF), který ji předá (S) k finálnímu zpracování.



Obr. 4.5 - Přenos RR paketů od koncových uživatelů



Obr. 4.6 - Přenos zpráv přes hierarchický strom

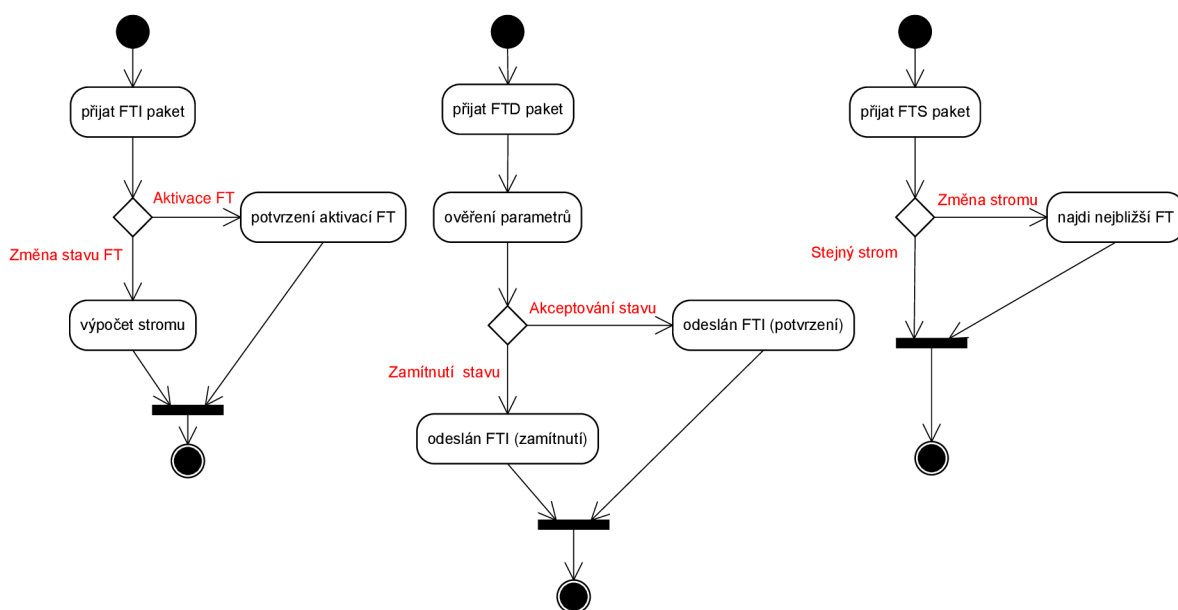
## 4.3 Stavové diagramy

Tato kapitola je věnována popisu změny stavu prvků v síti po příjmu zpráv FTI,FTD a FTS. Každý člen v síti reaguje jen na určité druhy zpráv. Dále je pak zjištěno jaký druh informace paket přenáší a provede podle toho definovanou akci. Přejchod mezi stavy zobrazují Obr. 4.7, Obr. 4.8, Obr. 4.9 a Obr. 4.10.

### 4.3.1 Zpracování paketu

Jak bylo probráno již v kapitole 2.5, každá zpráva protokolu TTP nese specifickou informaci pro zajištění funkčnosti celého zpětného kanálu. FTS přenáší stav hierarchického stromu, FTD definuje vlastnosti sumarizačního uzlu a FTI přenáší informace o správném či nesprávném fungování celé relace, či nutnosti provedení určitých změn v hierarchickém stromu. Pakety jsou jasně specifikovány a mohou se posílat jen určitým směrem, čímž se celá realizace protokolu hodně zjednodušuje.

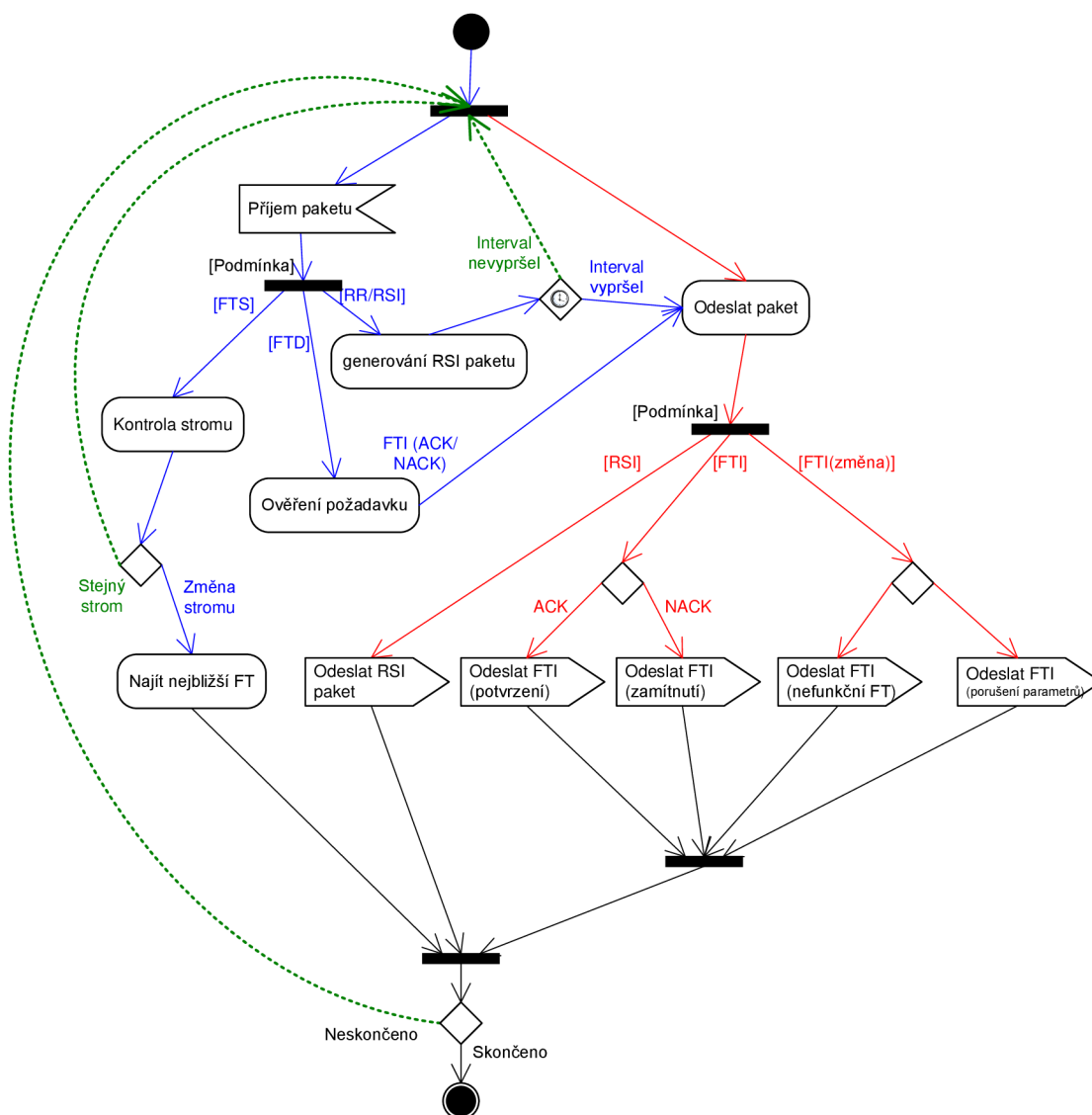
FTS paket je pak odesílám pouze (FTM) a obsahuje mapu vytvořeného zpětného kanálu s popisem parametrů. (FT) nebo (R) podle něho kontrolují stálost stromu a hledají nejbližší cíl pro odesílání zpráv. FTD paket má v zásadě velice jednoduchou roli. Je přijat vždy (FT) a nese informaci o parametrech, které (FTM) požaduje od (FT) dodržovat. FTI je pak posílán jen (FT) a obsahuje informaci buď o potvrzení či zamítnutí stavu anebo informaci o tom, že definované parametry už nezvládá dodržovat a je třeba optimalizovat hierarchický strom.



Obr. 4.7 – Zpracování paketů FTI,FTD a FTS po jejich příjmu

### 4.3.2 Stavový diagram FT

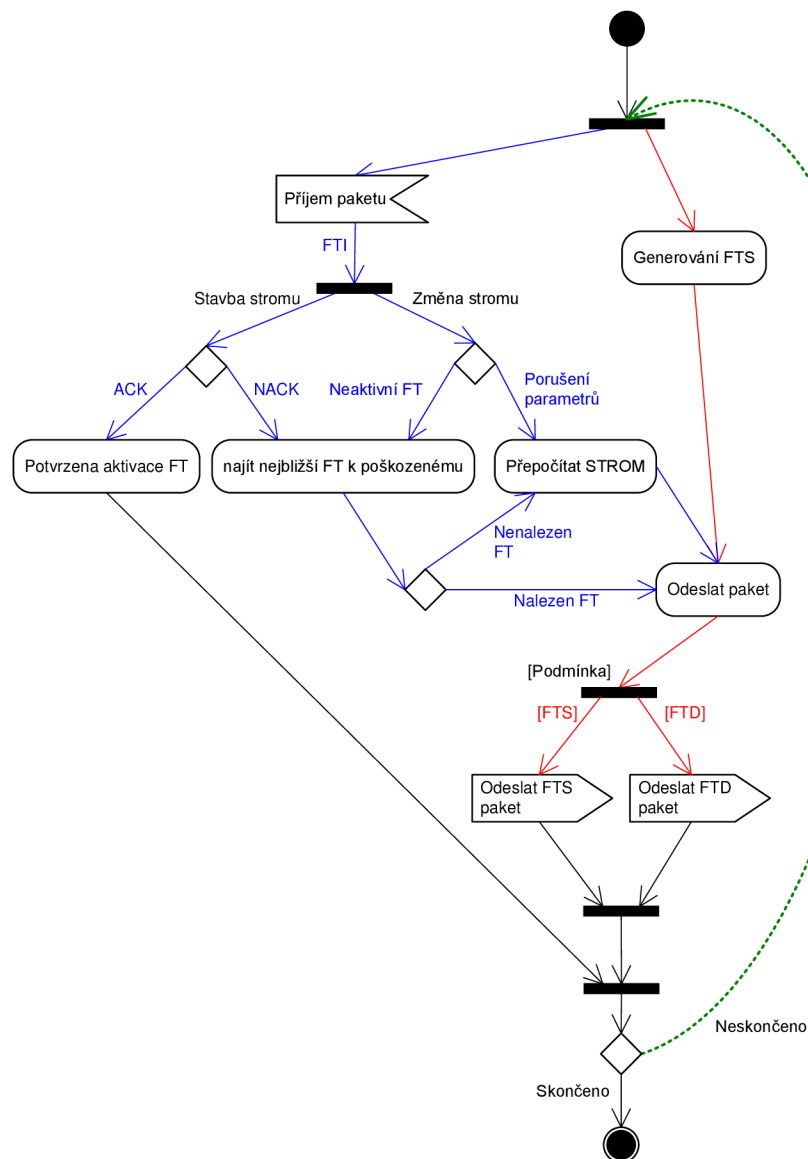
Na Obr. 4.8 je vidět kompletní provázanost procesů každého (FT). Jak již bylo ukázáno na Obr. 4.2, každý (FT) disponuje 3 vlákny pro vzájemnou komunikaci. Multicastové vlákno přijímá FTS pakety, z kterých kontroluje aktuálnost stavu stromu. Přijímací vlákno, na kterém (FT) přijímá RR/RSI pakety a FTD pakety (zprávy o kvalitě a definice stavu (FT)) a v neposlední řadě vysílací vlákno, které se stará o vysílání RSI paktů a FTI zpráv. (FT) přijímá 3 druhy paketů. Pokud přijme FTS, zkontroluje aktuální stav stromu, a buď ponechá aktuální (FT) pro odesílání RSI nebo nalezne nový. Pokud přijme RR/RSI zprávu, uloží její obsah do sumarizovaného RSI paketu a čeká na vypršení intervalu k jeho odeslání do další vrstvy. Pokud přijme FTD paket, zjistí obsah žádosti a pokud dané parametry může splnit, odešle paket FTI (potvrzení) nebo dá FTM vědět odmítnutí zprávou FTI (zamítnutí).



Obr. 4.8 - Stavový diagram FT

### 4.3.3 Stavový diagram FTM

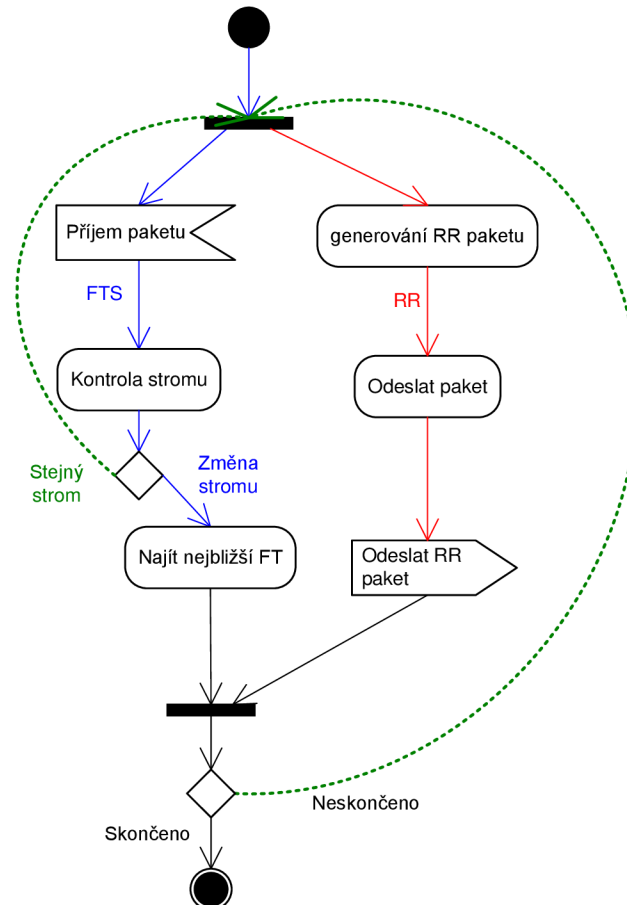
Na Obr. 4.9 je nakreslen stavový diagram (FTM), který realizuje přechod mezi jeho jednotlivými stavy. Jak už bylo několikrát zmíněno, (FTM) představuje hlavní jednotku, která je zodpovědná za realizaci zpětného kanálu TTP protokolu. Zde je možná překvapením, že je o něco jednodušší než stavový diagram (FT). Jeho role je totiž značně jednotvárná a sestává se především z výpočtu optimálního hierarchického stromu z dané množiny dostupných (FT) na základě příjmu FTI paketů. Podle nich poté zjišťuje, zda se jedná o fázi stavby stromu, anebo o pouhý monitoring a následnou reakci na podměty od (FT). (FTM) může řešit výpadek libovolného uzlu naleznutím jeho náhrady ve stejné oblasti, aby nemusel měnit celou strukturu zpětného kanálu. Nahradí tedy pouze jeden nefunkční (FT) za jeden funkční. Informace o kvalitě zpráv nebudou v případě výpadku krátkou chvílí chodit od jednoho (FT), ale (FTM) zajistí včasnou optimální náhradu.



Obr. 4.9 - Stavový diagram FTM

### 4.3.4 Stavový diagram Příjímače

V případě přijímače se jedná o nejjednodušší stavový diagram ze všech. (R) posílá RR zprávy v definovaném čase a navíc kontroluje multicastový kanál, kde (FTM) přenáší informace o stavu zpětného kanálu ve formě FTS.

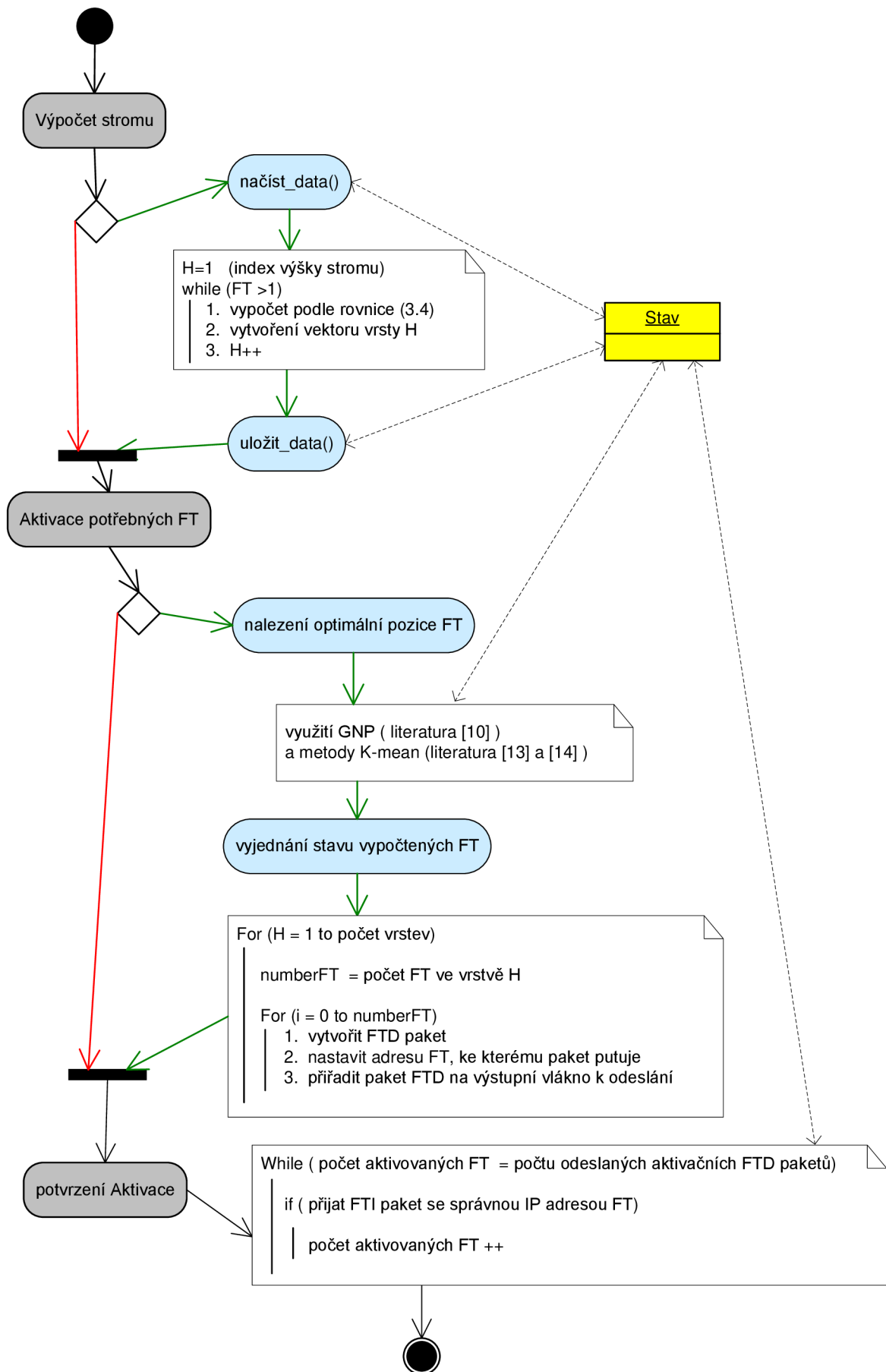


Obr. 4.10 - Stavový diagram Příjímače

### 4.3.5 Stavový diagram stavby hierarchického stromu

Na Obr. 4.11 je představen stavový diagram tvorby hierarchického stromu. (FTM) na základě příjmu zprávy FTI se žádostí o stavbu nového stromu načte potřebná data o aktuálním stavu stromu a podle Rovnice 3.4 zjistí aktuální počet vrstev nutný pro realizaci zpětného kanálu. Následně vybere z množiny (FT) takové, které nejlépe odpovídají algoritmu K-mean [12], [13] a vyšle k jim aktivační paket FTD. Definování (FT) na příjem paketu FTD reagují zasláním zprávy FTI, jak bylo ukázáno již na Obr. 4.8. (FTM) pak již jen čeká jaká odpověď o aktivaci dojde od tázaných (FT). Pokud by některý z (FT) odpověděl záporně nebo vůbec, (FTM) vybere za tento (FT) nejvhodnější (nejbližší) náhradu. Po dokončení tohoto procesu je již hierarchický strom připraven k použití a (FTM) začne vysílat do multicastového kanálu FTS zprávy.

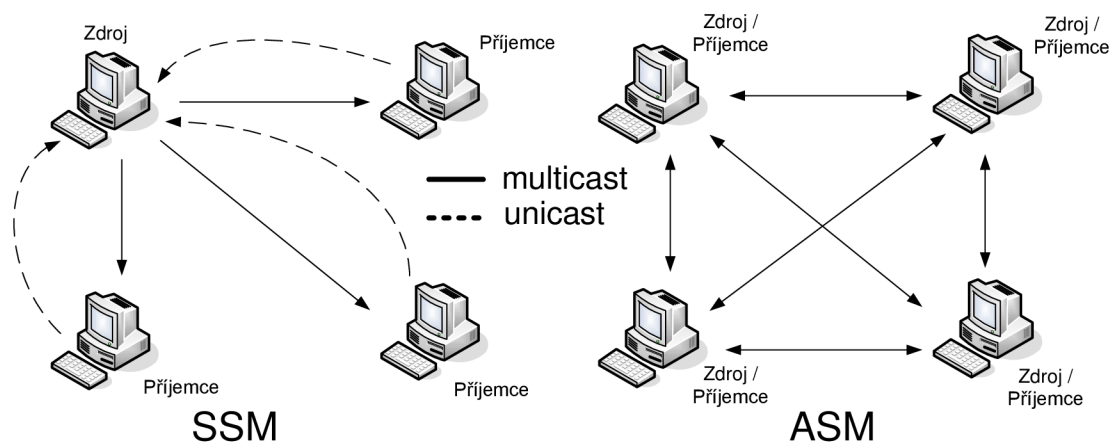




Obr. 4.11 - Diagram algoritmu sestavení H-vrstvého stromu

## 4.4 Řešení potíží při vývoji

V průběhu vývoje protokolu narážíme na stále nové potíže a problémy, které je třeba řešit. V úplných počátcích bylo třeba vyřešit využití protokolu ICMP [8], který není programovacím jazykem Java [20] vůbec podporován. V našem případě se jedná o vážný nedostatek, jelikož na odezvě příkazu „ping“ je postaven celý algoritmus pro výpočet vzdálenosti uživatele od pozičního bodu a následné nalezení pozice. Naštěstí existuje řada projektů v rámci open source license, které mají již tuto problematiku zvládnutu a zpracovány<sup>3</sup>. Dalším úskalím se stala podpora Source-Specific Multicast (SSM) [14] a [15]. Jedná se o typ multicastové komunikace one-to-many, kde je zpětný kanál řešen pomocí unicastu. Jen jeden člen v síti funguje jako zdroj dat a ostatní pracují jako příjemci. Síť PlanetLabu totiž podporuje zatím pouze Any-Source Multicast (ASM) [15], kde síť funguje na principu many-to-many a každý člen je zároveň zdroj i příjemce dat. Obr. 4.12 demonstruje příklad právě takového typu komunikace. Potíž je však v tom, že Java v současné verzi zatím nepodporuje SSM<sup>4</sup> a navíc pro správnou funkci SSM je potřeba aktualizovat operačního systému směrovačů pro podporu protokolu IGMPv3 [4], který v současné době není příliš rozšířen. V reálných sítích vzniká určité napětí mezi poskytovateli internetového připojení (ISP) a uživateli. Uživatelé požadují nasazení SSM multicastu s protokolem IGMPv3 a ISP na druhou stranu chtějí záruky, že jejich investice do směrovacích prvků, které tyto parametry bude splňovat, se jim vyplatí. Problém spojený se simulacemi řešíme pomocí simulační knihovny JSimLib3 popsané v kapitole 4.1.



Obr. 4.12 – Druhy multicastového vysílání

<sup>3</sup> <http://pdb.finkproject.org/pdb/package.php/jicmp> nebo <http://www.koders.com/java/fid93EB3022C6BCB6EAC082BE8CB13E925DB2F0266B.aspx>

<sup>4</sup> Podpora SSM je plánována pro Java v7.0, existuje ale i projekt JSSM, který přidává možnosti SSM do starších verzí Javy (autor: Radim Burget, <https://jssm.dev.java.net/>)

## 4.5 Volba mechanismu pro tvorbu zpětného kanálu

Přístupů k realizaci optimálního zpětného kanálu je hned několik, jak naznačuje kapitola 3. Vždy záleží na úhlu pohledu a na prioritách, podle kterých je struktura stromu koncipována. Důležitou roli hraje především velikost relace, spolehlivost a účinnost sumarizačních bodů, dále pak jejich počet a hlavně požadované parametry, které jsou hlavním omezením. V zásadě však existují 3 hlavní směry:

- Sestavení plně **asynchronního** zpětného kanálu **s dodržáním** parametrů RTCP
- Sestavení plně **asynchronního** zpětného kanálu **bez dodržení** parametrů RTCP
- Sestavení **synchronního** zpětného kanálu **s dodržáním** parametrů RTCP u přístupové vrstvy

### 4.5.1 Požadavky zákazníka

Jak bylo naznačeno již v úvodu této kapitoly, uživatel je hlavní důvod toho, proč je kladen takový důraz na kvalitu služby. Abych byl přesnější, v našem případě se jedná o jednoho uživatele, ale až o desítky a stovky tisíc koncových zákazníků. IPTV je služka, které poskytuje zábavu mnoha lidem, kteří jsou ve stejnou dobu na různých místech planety a používají nejrůznější možnosti připojení k internetu a také nejrůznější zařízení, na kterém IPTV sledují. Proto musí být TTP protokol natolik robustní aby dokázal vyhovět všem bez výjimky.

Každý zákazník má v zásadě 2 hlavní požadavky. Jedná se o:

- **Kvalitu**
- **Cenu**

Míra kvality je přímo neúměrná ceně, za kterou se daná služba provozuje. K tomu, aby tato propast byla co nejmenší, přispívá právě TTP protokol a tvorba optimalizovaného zpětného kanálu.

### **4.5.2 Požadavky a možnosti poskytovatele IPTV služby**

Poskytovatelé jsou vždy tlačeni zákazníky k tomu, aby poskytovali stále lepší a hodnotnější služby. Na trhu totiž existuje vždy druhý hráč, který by z eventuelní neschopnosti mohl profitovat a přebrat na sebe větší či menší portfolio zákazníků. Proto je třeba služby neustále vylepšovat, ale za cenu co možná nejmenších investic. Přesně takovou možnost protokol TTP přináší. Představuje softwarové řešení k tomu jak vyřešit limity spojené se šířkou pásma. Vždy existuje možnost zakoupení nové výpočetní kapacity, avšak to je spojeno s následnou mnohem větší režií a údržbou. Softwarové řešení přináší lepší alternativu, jelikož poskytovatelé mohou využít stávajícího zařízení a i samotná implementace je o mnoho jednodušší.

TTP protokol stojí někde na pomezí mezi hardwarovým a softwarovým vylepšením, i když se z větší části jedná o softwarové řešení. Nezbytnou podmínkou je totiž nakoupení několika nových serverů, které je nutné rozmístit do vhodných lokalit. Tento proces by se dal velice dobře přirovnat stavbě a plánování rozmístění BTS stanic v síti GSM. Poskytovatel má však i možnost pronájmu výpočetní kapacity od jiného poskytovatel internetových služeb (ISP), místo nákupu a provozování vlastních stanic. Tím vzniká naprosto různorodá síť, která spolu musí spolupracovat a komunikovat.

### **4.5.3 Asynchronní kanál s/bez parametrů RTCP**

Asynchronní kanál má tu výhodu, že ho není třeba nějakým způsobem synchronizovat. Interval odesílání zpráv je v celém zpětném kanále naprosto stejný a lze jej vypočítat z Rovnice 3.10 a Rovnice 3.9. Volbu tohoto režimu bude představovat asi většina realizací. Mechanismus správy hierarchického stromu je poměrně jednoduchý a přitom poskytuje velice kvalitní službu při nižším počtu uživatelů (do  $10^5$ , více Obr. 3.1). V současné době nelze ani předpokládat, že by počet uživatelů IPTV byl vyšší. Jedná se o poměrně novou službu a stále existuje spíše na okraji zájmu uživatelů. Za 10 či 20 let může být situace ale naprosto jiná a proto je protokol TTP koncipován tak, aby se dokázal vhodně přizpůsobit. Pokud nahlédneme zpět na Obr. 3.3, bude nám zřejmé, že počet (FT) u této metody je naprosto nejmenší, což uvítají i poskytovatelé IPTV služby.

Další důležitou vlastnost představuje dodržení parametrů RTCP protokolu. Jedná se především o interval odesílání RR zpráv od (R). Protokol RTCP definuje tento interval v rozmezí  $<5;\infty$  [1]. Tato podmínka je velice důležitá pro všechny stávající aplikace, které jsou pro tento interval optimalizovány. Pro každá taková program se v zásadě mění jen jedna věc a tou je cíl zpětné vazby. Už nebudou své RR zprávy odesílat přímo (S), ale naleznou pro sebe nejbližší (FT), který převezme jeho roli.

Na druhou stranu existuje oblast zájmu, kde požadujeme rychlou odezvu a nezáleží nám příliš na počtu sumarizačních bodů. Takovou oblastí může být například hraní online her, kde existuje řada serverů s nízkým využitím, jejichž potenciál se dá využít na zkvalitnění služby sběru informací od skupiny hráčů. TTP protokol lze totiž využít všude tam, kde potřebujeme přenést určitý druh informace od velkého množství uživatelů. Poté záleží na specifickém druhu aplikace, který požaduje buď dodržení parametrů RTCP nebo ne.

#### **4.5.4 Synchronní kanál**

Synchronní kanál představuje variantu především pro poskytovatele, kteří mají k dispozici dostatečné množství (FT) a volbou asynchronního kanálu by řadu z nich vůbec nevyužili. Synchronní kanál (Rovnice 3.9) umožňuje totiž snížit interval posílání RSI zpráv mezi vrstvami hierarchického stromu při zachování intervalu u přístupové vrstvy definovaného podle RTCP. Podmínkou pro správnou funkci celé struktury je synchronizace jednotlivých (FT) ve všech vrstvách vytvořeného stromu. Pokud budou intervaly odesílání ve vrstvách (FT) na sebe navazovat, jak bylo ukázáno na Obr. 3.4, lze dosáhnout nejen rychlejšího průchodu zpráv zpětným kanálem, ale navíc i získat stabilnější a objektivnější informaci od všech účastníků, jak bylo ukázáno na Obr. 3.5. Vzájemná synchronizace (FT) ve vybudovaném zpětném kanále je velice zajímavé téma, které však v rámci této práce není řešeno. Jedná se o podmět k dalšímu zkoumání, kterému bych se chtěl osobně v budoucnu dále věnovat. Synchronizaci lze řešit novým paketem, který by převzal tuto schopnost anebo v rámci FTI paketů, kdy by došlo k definici nových parametrů, které FTI paket bude obsahovat. V každém případě je třeba tomuto tématu věnovat více času a ověřit, do jaké míry synchronizační zprávy zatíží kanál a jak velkou výhodu v porovnání s asynchronním kanálem přinese.

## ZÁVĚR

Cílem diplomové práce bylo nalezení optimálního řešení sestavení hierarchického stromu zpětného kanálu pro sběr dat od velkého množství uživatel. Byla prakticky realizována kompletní sada paketů, které dohromady zajišťují provoz celé struktury. Praktická realizace je uskutečněna v programovacím jazyku Java a pakety lze najít na přiloženém CD. Jedná se o balíček s názvem „cz.feec.vutbr.packets.ttp“ a obsahuje všechny vstupní i výstupní metody, které umožňují s pakety plně manipulovat. Další částí je samotný mechanismus sestavení zpětného kanálu a jeho následná správa. V této práci byly představeny mechanismy, které demonstrují celou filozofii sestavení zpětného kanálu a stavy, které je nutné zvládnout, aby bylo dosaženo finálního výsledku. Pro přehlednost a vyšší porozumění celé problematice bylo na vysvětlení principů použito stavových a sekvenčních diagramů v notaci jazyka UML. Kód samotného mechanismu sestavení hierarchického stromu pro 10 (FT), 1 (RFT) a 1 (FTM) je opět přiložen na CD v balíčku „cz.feec.vutbr.sim.ha“. Nutno podotknout, že na úpravách se neustále pracuje, a tudíž se jedná pouze o stav práce, která byla k dispozici dne 20. 5. 2008. K dokončení celého projektu chybí ještě nemalý kus cesty, ale díky této práci máme k dispozici dostatečně detailní popis stavů na to, abychom byli schopni v brzké době přijít s neoficiální funkční verzí TTP protokolu. Zatím vše nasvědčuje tomu, že cíle definované na začátku projektu se podaří naplnit. Jedná se hlavně o přenos RR zpráv od obrovského počtu uživatelů k vysílači multimediálního obsahu v co nejkratším čase. Jak bylo prezentováno v této práci, jedná se v ideálním případě až o 300 násobné snížení intervalu příchodu těchto zpráv. V podmínkách reálné sítě bude tento násobek jistě menší, ale i tak se stále bude jednat o obrovské zlepšení stávající architektury postavené na protokolu RTP/RTCP.

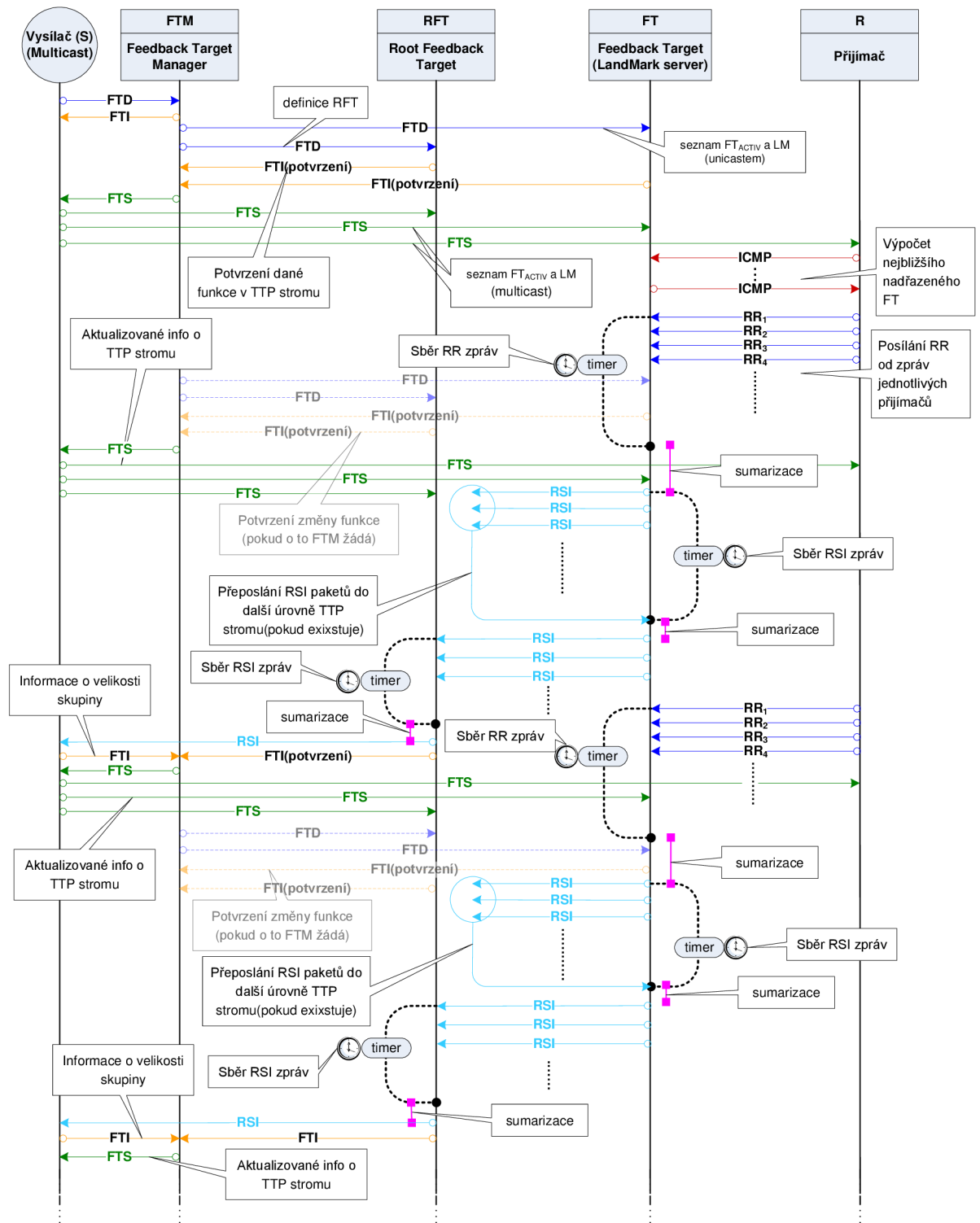
## Literatura:

- [1] H. SCHULZRINNE, S. L. CASNER, R. FREDERICK, V. JACOBSON, *RTP: A Transport Protocol for Real-Time Applications*, RFC 3550 , Internet Engineering Task Force, červenec 2003 <<http://www.ietf.org/rfc/rfc3550.txt>> [online červen 2008]
- [2] J. OTT, J. CHESTERFIELD, E. SCHOOLER, *RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback*, Internet draft-ietf-avt-rtcpssm-14, Internet Engineering Task Force, 16 Listopadu 2007, <<http://www.ietf.org/internet-drafts/draft-ietf-avt-rtcpssm-14.txt>> [online červen 2008]
- [3] S. RATNASAMY, M. HANDLEY, R. KARP, S. SHENKER, *Topologically-Aware Overlay Construction and Server Selection*, Proceedings of 21<sup>st</sup> Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, 2002.
- [4] B. CAIN, S. DEERING, I. KOUVELAS, B. FENNER, A. THYAGARAJAN, *Internet Group Management Protocol, Version 3*, RFC 3376, Internet Engineering Task Force, říjen 2002 <<http://www.ietf.org/rfc/rfc3376.txt>> [online červen 2008]
- [5] D. KOMOSNÝ, V. NOVOTNÝ, *Feedback Distribution in Specific Source Multicast using Tree Transmission Protocol*, in ICN07 - The Sixth International Conference on Networking . Martinique, 2007, ISBN 0-7695-2805-8
- [6] R. BURGET, D. KOMOSNÝ, *Real-time control protocol and its improvements for Internet Protocol Television*, International Transaction on Computer Science and Engineering, ISSN 1738-6438, 2006, 2006, č. 31, s. 1 - 12.
- [7] D. KOMOSNÝ, V. NOVOTNÝ, *Analysis of bandwidth redistribution algorithm for single source multicast*, In Proceedings of the Sixth International Network Conference. Sixth International Network Conference. United Kingdom: University of Plymouth, 2006, s. 45 - 52, ISBN 1-84102-157-1
- [8] J. POSTEL, *Internet Control Message Protocol*, Internet Engineering Task Force, září 1981 <<http://www.ietf.org/rfc/rfc792.txt>> [online červen 2008]
- [9] T. S. EUGENE NG, HUI ZHANG, *Towards Global Network Positioning*, Extended Abstract, ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, listopad 2001
- [10] F. DABEK, R. COX, F. KAASHOEK, R. MORRIS, *Vivaldi: A decentralized network coordinate system*, Proceedings of the ACM SIGCOMM, Portland, OR, srpen 2004.
- [11] PUNEET SHARMA, ZHICHEN XU, SUJATA BANERJEE, SUNG-JU LEE, *Estimating network proximity and latency*, ACM SIGCOMM Computer Communication Review, v.36 n.3, červenec 2006

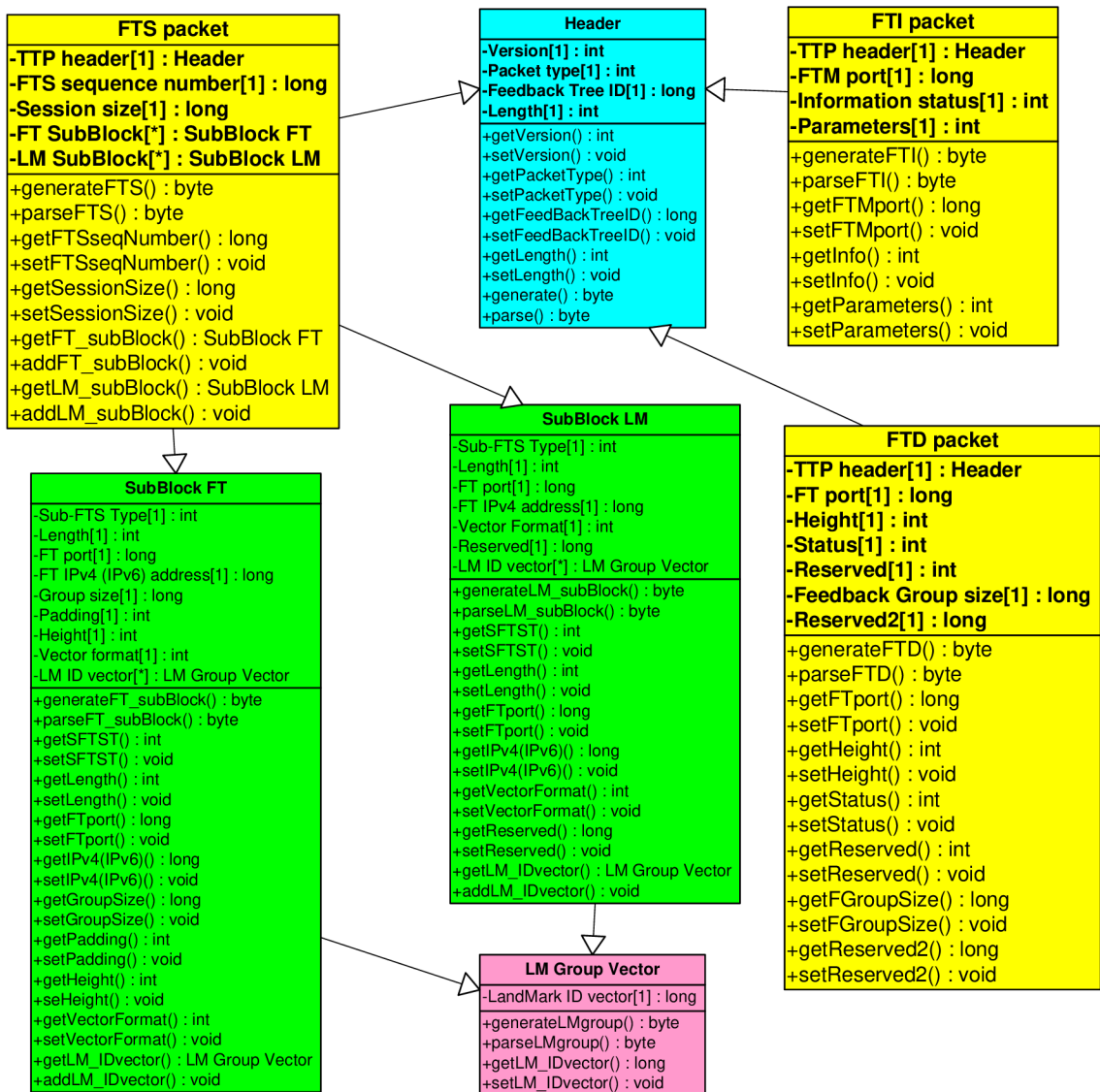
- [12] J. A. HARTIGAN , M. A. WONG, *A K-Means Clustering Algorithm*, Applied Statistics, v.28 , 1979
- [13] Wikipedia - The free encyclopedia, *K-mean Algorithm*, <[http://en.wikipedia.org/wiki/K-means\\_algorithm](http://en.wikipedia.org/wiki/K-means_algorithm)>, [online červen 2008]
- [14] BHATTACHARYYA, S., *An Overview of Source-Specific Multicast (SSM)*, RFC 3569, Internet Engineering Task Force, červenec 2003, <<http://www.ietf.org/rfc/rfc3569.txt>>, [online červen 2008]
- [15] B. QUINN, K. ALMERTH, *IP Multicast Applications: Challenges and Solutions*, RFC 3170, Internet Engineering Task Force, září 2001, <<http://www.ietf.org/rfc/rfc3170.txt>>, [online červen 2008]
- [16] Object Management Group , *Unified Modeling Language Specification*, <<http://www.uml.org/>>, [online červen 2008]
- [17] Wikipedia - The free encyclopedia, *Global Positioning System*, <[http://en.wikipedia.org/wiki/Global\\_Positioning\\_System](http://en.wikipedia.org/wiki/Global_Positioning_System)>, [online červen 2008]
- [18] Wikipedia - The free encyclopedia, *Triangulation*, <<http://en.wikipedia.org/wiki/Triangulation>>, [online červen 2008]
- [19] Wikipedia - The free encyclopedia, *Internet Protocol Television*, <<http://en.wikipedia.org/wiki/IPTV>>, [online červen 2008]
- [20] Sun Microsystems, *JDK™ 6 Documentation*, <<http://java.sun.com/javase/6/docs/>>, [online červen 2008]
- [21] PlanetLab, „About“, < <http://www.planet-lab.org/about>>, [online červen 2008]



# Příloha:



Příloha 1 - Sekvenční diagram provozu ve zpětném kanále



Příloha 2 - Kompletní diagram tříd TTP paketů

## Obsah Příloženého CD:

- Diplomová práce ve formátu pdf a docx + metadata
- Veškeré UML diagram prezentované v práci (*Stavový diagram, Diagram Tříd paketů, Sekvenční diagram,...*)
- Všechny zdrojové M-File ze simulací prováděných v prostředí Matlabu
- Balíček „*cz.feec.vutbr.packets.ttp*“ s kompletní realizací všech TTP paketů
- Balíček „*cz.feec.vutbr.sim.ha*“ s praktickou realizací hierarchického stromu pro 10 (FT), 1 (RFT) a 1 (FTM)