Advantage of Hierarchical Aggregation

Jakub Muller, Dan Komosny, Radim Burget, Patrik Moravek

Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic

Summary

Running a service for millions of clients based on Real-Time Protocol/Real-Time Control Protocol (RTP/RTCP) entails a huge amount of data periodically generated by the clients. These data have to be transferred to the sender as soon as possible because of the quality measurement of the real-time service. The solution lies in the hierarchical aggregation and the summarization in a feedback channel. Based on these requirements a new type of a protocol will be presented in this paper.

Key words:

hierarchical aggregation, summarization, localization, feedback channel

1. Introduction

There has been a significant expansion of the client internet connection speed in recent years. This is why the customers are looking for new kinds of entertainment on the Internet. They are focusing on multimedia services like Video over IP, Voice over IP and also IP Television (IPTV). Especially in the case of IPTV, the number of clients can be enormous and the classical way of broadcasting the media content via RTP/RTCP [1] cases to provide the quality service. The RTP/RTCP bandwidth is divided into two parts. 95% is reserved for RTP forward channel and only 5% is reserved for information about quality of the session. This service is provided by Real-Time Control Protocol. In 5% of the session bandwidth you have to transfer all data that are periodically generated by clients. Each client sends its data directly to the sender and if there are millions of clients in one session, the periodical time for sending the information about quality can be hours not seconds as required by quality of service. Because of this we have invented a new type of feedback channel and it will be presented in this paper.

1.1 Key ideas of new strategy

All our ideas are implemented in new type of protocol called Tree Transmission Protocol (TTP) [2]. It has been designed and developed at University of Technology in Brno, Department of Telecommunications (Czech Republic). It implements methods of hierarchical aggregation, summarization and localization to provide the capability of transferring a huge amount of data via the narrow channel. This protocol can be used anywhere where we have similar data on the client side and we need to transfer them via the channel as fast as possible. This situation is represented by IPTV but there are also others like sensor network architecture. Let us imagine sensors measuring the quality of water, air pollution, and pressure and so on. The sensors communicate via wireless channel which cannot offer wide and fast connection and the amount of data can be enormous. There is also much redundancy in the data. This is the way we use method of summarization to reduce this redundancy and save the necessary bandwidth. This type of process is made by summarizations nodes. There is also need for localization because there can be many the summarization nodes in the feedback channel and each client has to send its data to the closest one.

2. Tree Transmission Protocol (TTP)

The TTP protocol is based on cooperation between the manager of summarization nodes, called Feedback Target Manager (FTM), summarization nodes themselves, called Feedback Targets (FT), and clients. All these members together create a kind of tree in the feedback channel as you can see in Figure 1. FTM is not a physical part of the tree because it only manages what the tree will look like and what kind of parameters will be carried on in the nodes.

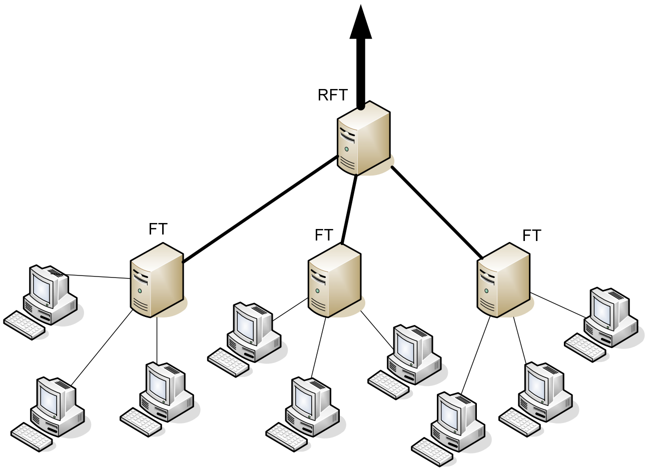


Figure 1 - Structure of feedback channel

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

In Figure 1 you can see the diagram of the feedback channel and in Figure 2 is depicted the flat model of the feedback tree on the map of Europe is depicted. Here you can see how important the localization of the members is. There is no good reason for the client in Russia to send its data to FT located in Spain if there is a closer FT located in Russia. We need to find the shortest path to the sender to provide the shortest time of data propagation. Localization also divides all clients into groups which are highlighted in circles. Each FT thus provides summarization only for clients in its closest area.

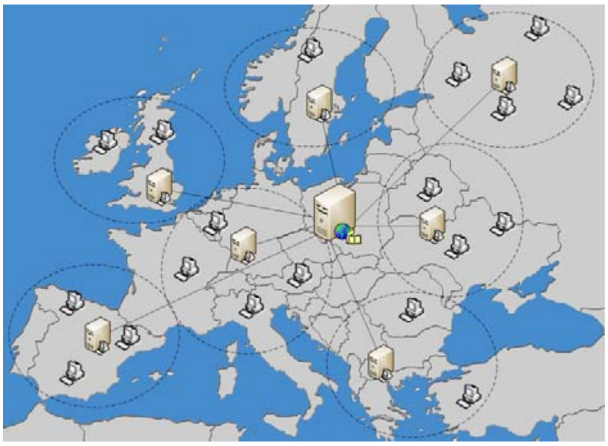
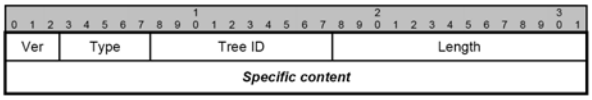


Figure 2 - Localization of members

2.1 Packets

We have created several types of packets which are used in the case of TTP for creating and managing the feedback channel. You can see the general header of all packets in Figure 3. The specific content is dependent on the type of packet. We have reserved 5bits for the definition of the packet type. It gives us an opportunity to use up to 32 types of packets in the case of a TTP protocol. Another important field is the "Tree ID". FTM or FT can run more trees and FT must be able to recognize witch tree the packet belongs to. The maximum number of tree feedback channels is set to 1024. The length field defines the size of the received packet.

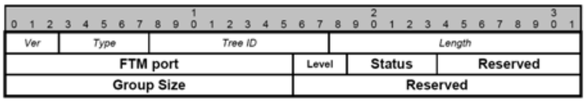


Ver	Type	Tree ID	Length
Specific content			

Figure 3 - General header of TTP protocol

2.1.1 Feedback target definition packet (FTD)

FTD is generated by FTM and it is used for specifying the parameters on FT. It is sent via TCP to the IP address of defined FT. FT needs certain time for verifying the information in the FTD packet and for generating the response (see FTI packet, chapter 2.1.2), which contains information whether the FT is able to offer these parameters. You can see specific fields of FTD packet in Figure 4. Certain fixed parameters are defined here and some fields are reserved for the future expansion. The most important from the fixed parameters are "Level", "Status" and "Group size". "Level" is used for defining level of the feedback tree where FT will be operating. "Status" defines special parameters, e.g. if FT will be Root FT (the highest FT in the tree), Landmark (will serve as localization point, charter 2.2) and others. "Group Size" defines how many clients can communicate with the defined FT. If this number is exceeded the given FT sends the FTI packet (see chapter 2.1.2) with defined issue to FTM and it optimizes the state of tree.

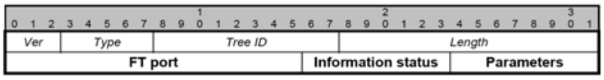


Ver	Type	Tree ID	Length
FTM port	Level	Status	Reserved
Group Size	Reserved		

Figure 4 - FTD packet

2.1.2 Feedback target information packet (FTI)

This packet is mostly sent by FT, but it can be also sent by FTM as information about removing defined FT from the feedback tree. FTs send this packet to FTM as a response to FTD packet to FTM and it contains either confirmation or refusal of information about received parameters. It is also sent by FT as warning information that some parameters are respected no more or when some FT does not receive data from another FT in lower level of tree. You can see the defined structure of FTI packet in the Figure 5.



Ver	Type	Tree ID	Length
FT port	Information status	Parameters	

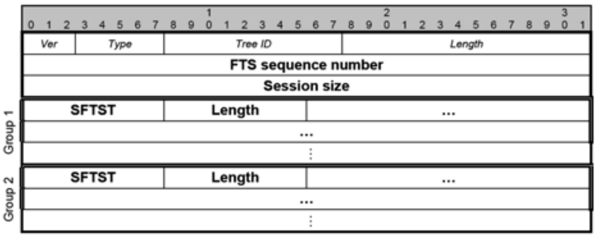
Figure 5 - FTI packet

2.1.3 Feedback target specification packet (FTS)

FTS is generated by FTM and it is transferred to all members of session (FTs and all clients) via the multicast channel. This packet contains information about all FTs and about their parameters and location. Reading this packet, end users and FTs are able to find its closes summarization point and start to send data to this destination. The size of FTS packet depends on the number of FTs in the feedback tree. The number of FTs could be a huge number and FTS need to be sent periodically because new members can join session at any time. It is like a "teletext" service on your classical analogue TV. All data in FTS packet should be sent in about 5 second intervals. You can find many so called SubBlocks in each FTS. This SubBlock contains all necessary information about the defined FT. There are two types of SubBlocks. The first defines the parameters of FT (Figure 7) and the second defines the parameters of LandMark (Figure 8, chapter 2.2). The structure of FTS packet is depicted in Figure 6.

The main fields of FTS packet are:

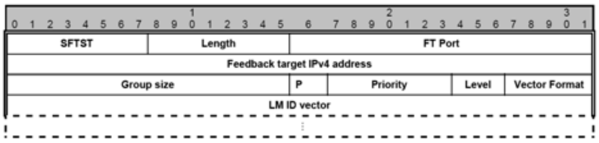
- **FTS sequence number** (32bits) for the packet number
- **Session size** (32bits) for actual number of joined clients (max. $2^{32} \approx$ 4,3 billion clients)
- **SFTST** type of SubBlock



	Ver	Type	Tree ID	Length
	FTS sequence number			
	Session size			
Group 1	SFTST	Length	...	
	...			
	⋮			
Group 2	SFTST	Length	...	
	...			
	⋮			

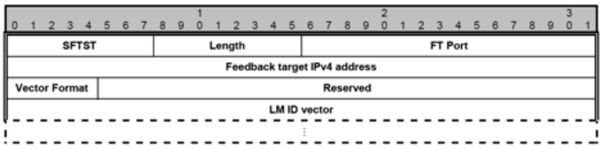
Figure 6 - FTS packet

FT SubBlock (Feedback SubBlock) defines the parameters of selected FT and contains also data (vector) about its position. The current version of this SubBlock is depicted in Figure 7. The number of vectors is flexible as you can see. It gives us the opportunity to use different types of localization methods and also different types of dimensions that we can use for better location.



SFTST	Length	FT Port		
Feedback target IPv4 address				
Group size	P	Priority	Level	Vector Format
LM ID vector				
⋮				

Figure 7 - Feedback target SubBlock



SFTST	Length	FT Port
Feedback target IPv4 address		
Vector Format	Reserved	
LM ID vector		
⋮		

Figure 8 - Landmark SubBlock

LM SubBlock (LandMark SubBlocks) is very similar to the previous one. It contains basic information about the LandMark server (more in the next chapter) and most importantly about its position. The structure is depicted in Figure 8.

2.2 Localization server (Landmark - LM)

LandMark is a well-known server for the whole session [3]. This server knows its position, which is stable. All members contact about 5 such servers to find out how far they approximately are and calculate their position. We need this distance from the client because it needs to select the closest FT in the whole session. There could be also a problem with contacting the LM, because each member generates about 10 requests for one LM and in the case of millions of members this can cause the overloading. We need to test this situation first but we suppose that this should not be a problem, because the request messages are not generated at the same time to the same LM in real network.

Figure 9 shows two PCs and two nodes creating their vector table. PC marked as PC_2 is somewhere in Italy and it performs the localization process with landmarks A, B, C and D and it produces a table of landmark vectors (Table 1). All other members perform the same process and produce similar tables. Thanks to this table and the fix location of all LMs, they are able to calculate their position. Many projects deal with this situation using the GNP [4] or Vivaldi [5] algorithm.

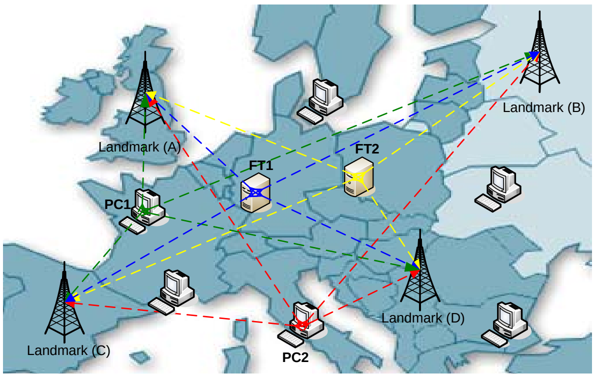


Figure 9 - Finding location using LM

Table 1- Table of PC2 localization values

	Landmark	Delay (ms)
1	C	85
2	D	79
3	A	120
4	B	250
5	…	…

2.3 Feedback Target (FT)

FT represents the 'worker' in the feedback tree. It summarizes data for clients in its group, like the sender does and creates the so-called RSI (Receiver Summary Information) packet [6], which is sent to the higher level of the tree. FT is able to work simultaneously in several trees with different tree IDs. FT can also work as LM if it has hardware capacity for this service. It operates on any level with other FTs. An exception is the last level; here is the single FT is called the Root Feedback Target (RFT). It does final summarization and it is unique in the tree.

2.4 Feedback Target Manager (FTM)

FTM represents the main part of the TTP protocol. It is familiar with all parameters of all FTs and can create the feedback tree on request of the sender. It also controls the function of the feedback trees and optimizes their function. FTM is also responsible for publishing the state of all trees via FTS packets.

2.4.1 Dividing into the groups („CLUSTERING“)

As we already said, each member in the session needs to know the closest FT for faster propagation. So-called landmarks are used for this purpose (chapter 2.2). First of all, FT needs to register with FTM and it sends the table of delay time as you can see in Table 1. That means that FTM has a table of delay time from all FTs in a network and it can create any kind of the feedback tree as it wants. It sends information about this tree in FTS packet to all members and then they are able to find the closest FT from the defined tree. At the end of the process, all clients are divided into groups. We call this process 'clustering' and the groups are called 'clusters' (more in Figure 2) [3] and is based on ICMP response from LMs. Each 'cluster' has a limitation of the client's capacity and therefore FTM must be able to create a new FT in the case of need.

2.4.2 Creation of tree in the network

If the sender needs to create a new tree for a new multimedia session, it asks FTM in the FTD packet and FTM creates a tree from registered FTs. You can see the sequence diagram in Figure 10. FTM chooses the defined number of FTs and tries to activate them using the FTD packet. Each FT finds out if it is capable of providing the requested service and sends the FTI packet (acceptance or rejection). If FTM activates the necessary number of FTs, it will send all their parameters in the FTS packet. FTS packet is sent periodically because a new client of multimedia session needs to find out the state of feedback tree as soon as possible.

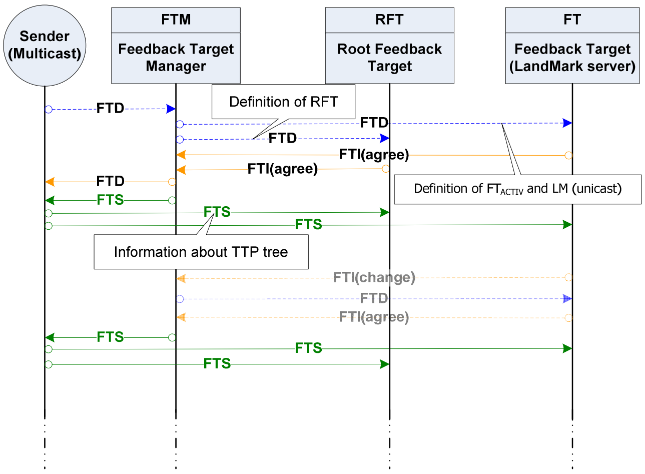


Figure 10 - Sequence diagram of creating new session

Each member of the session receives an FTS packet from the multicast channel and it is able to get from it IP address of the closest FT. In Figure 10 is also depicted a situation when some FT finds out a problem in its "cluster". Generally it is a situation when there are more members in the "cluster" than there should be. Thus, FT sends an FTI packet to inform FTM about the current state and FTM makes the appropriate changes. Figure 11 describes in detail how the tree is created.

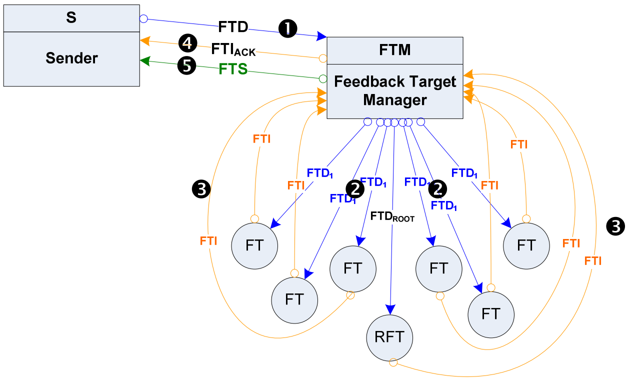


Figure 11 - Detailed process of the tree establishment

2.4 Clients

After the tree has been established, clients start sending their data about the quality of channel or other information back to the sender. They have to find the IP address of the closest FT from FTS packet. The client has to monitor the FTS packets because a change can occur in the tree at any time. The client also finds out from the FTS packet the number of members in the 'cluster' the mentioned client belongs to. It calculates the sending interval and knows how often the specific message can be sent. The propagation of the message takes certain time, but finally the message is received by the sender. You can see the process of message propagation in Figure 12 and Figure 13. Data are generated by the client and they are sent to the closest FT, which summarizes the messages from its "cluster" and forms an RSI packet [6]. This packet represents some kind of histogram, where the information from clients is stored. The RSI packet travels via all levels of the tree and is finally received and processed by the sender.

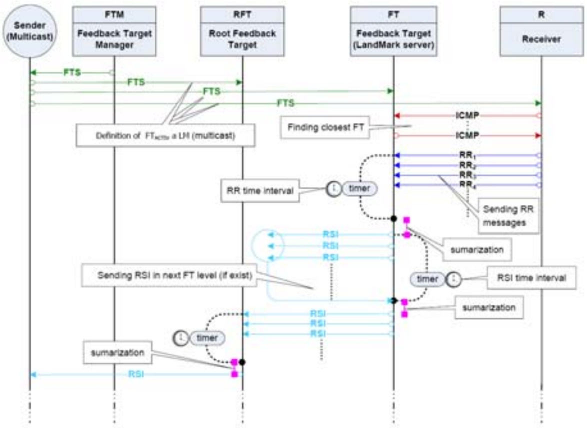


Figure 12 - Sequence diagram of data propagation

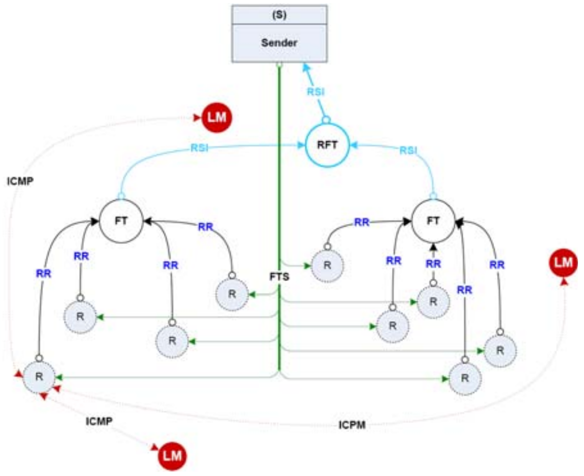


Figure 13 - Propagation of the messages in the feedback channel

3. Mathematical equations

Generally said, the TTP protocol represents a mechanism how to transfer a huge amount of data via a narrow channel. TTP creates several such channels and then summarizes data in FTs and gradually reduces the number of created channels until only one remains. These processes are represented by equations (1), (2) and (3). Let us start with the basic facts. The number of FTs on the first level of the tree (FT_1) can be calculated from the number of clients (N), size of transferred message (PL_1), interval for periodical transmission (T_1) and bandwidth of the feedback channel (BW_{FT}). There are also some limitations but they are not so important for us in this moment. It always depends on the type of the tree. Using these parameters, we obtain:

$$FT_1 = \frac{N \cdot PL_1}{T_1 \cdot BW_{FT}} \qquad [-] \qquad (1)$$

If we focus on the next level and substitute F_1 from equation 1, we obtain:

$$FT_2 = \frac{FT_1 \cdot PI_2}{T_2 \cdot BW_{FT}} = \frac{N \cdot PL_1 \cdot PL_2}{T_1 \cdot T_2 \cdot BW^2{}_{FT}} \qquad [-] \qquad (2)$$

The situation will be the same for the next levels and we can write generally for any level (H):

$$FT_H = \frac{N \cdot \coprod_1^H PL}{RW_{FT}^H \cdot \prod_1^H T} \qquad [-] \qquad (3)$$

3.1 Simulations

We would like to show you experimental results of the hierarchical structure in the feedback tree. We did certain theoretical simulations of an IPTV session in the MatLab environment, which confirm that TTP has a big potential. The values used in simulations below are the following:

$\boldsymbol{BW_{FT}}$ bandwidth used for transferring data from client to FT or from FT to FT on higher levels (140kb/s in simulations)

$\boldsymbol{PL_H}$ size of the packet on level H (from 2nd level RSI packet) (60b on the 1st level, 8kb on the next levels in simulations)

N number of clients in the whole session (1 000 00 in the simulation)

In the first graph (Figure 14) you can see a simulation which uses equation (1) and shows us how many clients can be joined in one session, having an interval for sending data set to at least 5 seconds. We can say that the number could be unlimited, but for real applications it has also certain limitations. The main limitation is the number of FTs ("dots" in Figure 15) and the number of levels in the feedback tree. But these limitations can be eliminated by choosing appropriate values in equation (3). This process is managed by FTM and therefore this mechanism is implemented here. In the second graph (Figure 14 you can also see the time propagation of message from one client with/without a hierarchical tree.

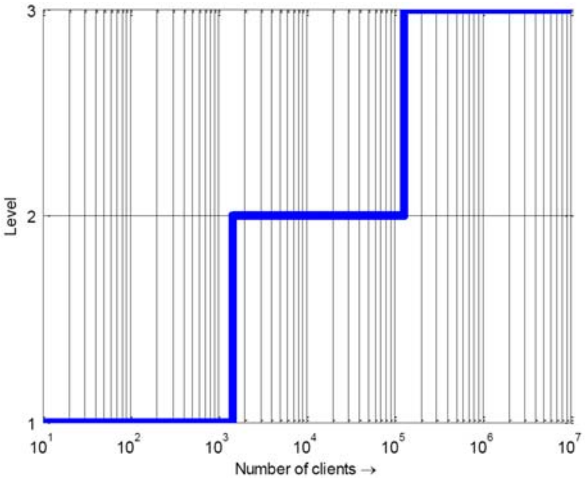


Figure 14 - Number of levels dependenced on number of clients

The values in Figure 15 can be interpreted as follows. If we have 1 000 000 clients with values, we need at least a 3-level tree according to equation (3). Each feedback tree ends with RFT (chapter 2.3) and this RFT is unique in the tree. Finding the level of a certain tree means calculating the number of FTs on each level until FTH in equation (3) is less than one. The green line shows the actual time of propagation on each level of the feedback tree.

4. Related work

HA is applied in several other technology branches and is closely related to many other projects. This section will mention several of the most important. Also some outcomes of work done by our team related to this work is referenced here.

A. Tree Transmission Protocol (TTP)

Tree Transmission Protocol (TTP) [2] is a protocol designed by our research team for HA. It is currently in the development stage. It is being modified based on the results of recent research. We hope that it will soon become the first realization ready to run in the commercial area.

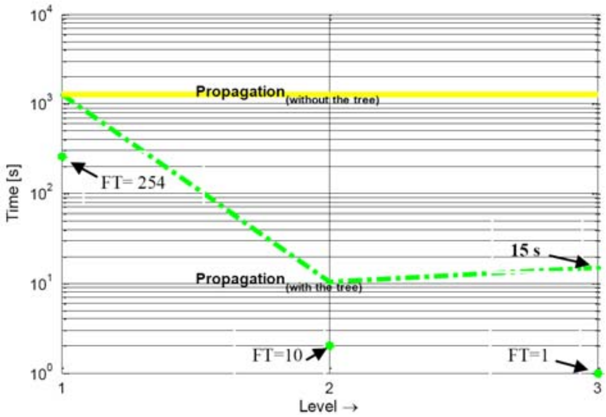


Figure 15 - Time propagation for 1 000 000 clients

B. RTCP extension for Single-Source Multicast Session with Unicast Feedback

RTCP extension [6] is optimization of RTP/RTCP [1] definition and using aggregation on sender side considerably reduces total amount of data sent by sender. In contrast to the other technologies described here, RTCP extension is still not capable of propagating signals from receivers at speeds comparable to those achieved with HA methods.

C. CPE WAN Management Protocol

CPE WAN Management Protocol [7] is a protocol designed for management of DSL networks. Its advantage is that it can easily pass through firewalls, as it uses HTTP protocol. Its disadvantage is that it uses less effective communication and is not designed for large-scale sessions. Thus its overhead is considerably higher than in methods using HA.

D. Simulations

As a part of this document, simulation of GNP [4], Vivaldi Coordinate [5] systems and TTF algorithm has been created. They were implemented in JAVA programming language and were published online on our web [8]. As they all are integrated with graphical user interface, they can be used for experiments and identification of potential problems.

E. Sensor network

Sensor networks are heralded as one of the most important technologies for the 21th century by Business Week. The HA in sense of receiver signaling and the methods discussed here can be extended to wireless sensor networks to gather receiver data in really short time and in an energy and time effective manner [9].

5. Conclusion and future direction

TTP protocol comes with a big potential. We have designed it for transmission of data in sessions with a huge number of clients and it can be use in many applications like IPTV, VoIP or even in sensors network. We have focused on improvement of IPTV so far but we are planning support in sensors network as a next step of development. As we have shown, TTP brings manageability and great control over the feedback channel even for millions of clients. None of the nowadays well known protocols can offer this kind of service. It is true that we have a long journey before us, but we are more than optimistic.

Acknowledgments

This work was supported by the Academy of Sciences of the Czech Republic, project 1ET301710508.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 3550: RTP: A Transport Protocol for Real-Time Applications. Technical report, IETF, 2003.

[2] Radim Burget Dan Komosny and Vit Novotny. Tree transmission protokol for feedback distribution in iptv systems. Communication Systems and Networks, 2008.

[3] R.Karp S.Shenker S.Ratnasamy, M.Handley. Topologically-aware overaly construction and server selection. In INFOCOM 2002. The 27th Conference on Computer Communications. IEEE, pages 1190– 1199, 2002.

[4] E. Ng and H. Zhang. Predicting internet network distance with coordiantes-based approaches, 2001.

[5] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. SIGCOMM Comput. Commun. Rev., 34(4):15–26, October 2004.

[6] J. Ott, J. Chesterfield, and E. Schooler. RTCP Extensions for Single- Source Multicast Sessions with Unicast Feedback. Technical report, IETF, 2004.

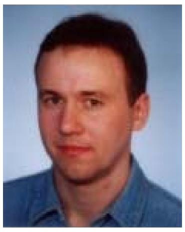
[7] DSL Home-Technical Working Group. Cpe wan management protocol, 2004.

[8] Radim Burget Dan Komosny. Ttf simulation. Technical report. [online], Multicast IPTV Research Group, 2008, http://adela.utko.feec.vutbr.cz/projects/tree-organizing.html

[9] Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient Communication Protocols for Wireless Microsensor Networks. In International Conference on System Sciences, Maui, HI, January 2000.

Ing. Jakub Müller, *(1984) received his M.Sc. degree in Electrical Engineering in 2008 and he is the PhD student and assistant at the Department of Telecommunications of Brno University of Technology in Czech Republic. He is engaged in research focused on IPTV, hierarchical aggregation and mobile application. He has experience with development of J2EE applications.

Ing. Radim Burget, *(1982) received his M.Sc. degree in Faculty of Information Technology in 2006 and he is the PhD student and assistant at the Department of Telecommunications of Brno University of Technology in Czech Republic. He is engaged in research focused on IPTV. He has experience with development of J2EE applications.



Ing. Dan Komosný, Ph.D., *(1976) received his Ph.D. degree in Teleinformatics in 2003. In 2003 he became an assistant professor at the Department of Telecommunications, Brno University of Technology. His research is aimed on multimedia systems; communication protocols; real-time communication; VoIP technology; IPTV technology, IP network quality of service and communication pragmatics for Wireless Sensor Network. He has practical experience with visual software development; real-time software development and formal description of communication protocols. He received or participated in several projects focused on networking research. He is a member of IEEE Communication Society.

Ing. Patrik Morávek, *(1984) received his M.Sc. degree in Electrical Engineering in 2008 and he is the PhD student and assistant at the Department of Telecommunications of Brno University of Technology in Czech Republic. He is engaged in research focused on WSNs . He has experience with development of C, C++ applications.