Scalable and Robust Active Element Network

Michal Procházka[1,2] and Miloš Liška[1,2]

[1] Faculty of Informatics,
Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
[2] CESNET z. s. p. o., Zikova 4, 160 00 Praha 6, Czech Republic
e-mail: xprocha7@fi.muni.cz, xliska@fi.muni.cz

Abstract. Modular user empowered reflector or Active Element (AE) is very useful application providing network support for building collaborative environment for groups with small to medium number of users. Single reflector has problems with robustness and scalability because of its centralistic architecture and thus the concept of stand-alone reflector has been extended to distributed Active Elements. Overlay network of AEs can get over these problems and besides this, it provides possibility to equip individual nodes with heterogeneous functionality. Similar to the reflector, we want the network of AEs to follow the user-empowered paradigm as well. Peer-to-peer (P2P) networks have the required properties for controlling such overlay network and some other which will be very useful for solving our problems. AE uses overlay network to transfer management data, which is for managing and recovering topology of data distribution, while the actual data is distributed over the host network. Our goal has been to design overlay network of AEs which is be robust, scalable, self-organizing, user empowered and pervasive.

1 Active Element

Active Element (AE) [1] is the main component of the whole system which we study and we would like to enhance. Briefly, AE is user-empowered software based on UDP packet reflector which replicates all incoming traffic to all registered clients. The term user-empowered means that the whole environment can be set up by user, there is no need of any network or system administrators. Because all data goes through the AE therefor the AE can process them in various ways. AE is element which allows to build user-empowered collaborative environment.

2 Network of Active Elements

The serious problem of the single AE is limited scalability and a missing robustness that is why an idea of network of AEs was proposed. The first solution was that AEs are interconnected by rules defined by the data distribution models. But there is a limitation of manual set up of each collaborative session.

We proposed self-organized AE network which would ensure organization of AEs in the network, dynamicaly creating collaborative sessions and besides this a higher level of robustness and scalability of the whole collaborative environment. The self-organized AE network should consists of two layers. The first is created by underlying network with raw data flow, in our case it is Internet. The second layer is overlay network which ensures self-organization of AEs, where we used P2P overlay network.

Actually the level of robustness and scalability of the AE network depends on three aspects. The first point where we can get more robustness and scalability is in the overlay network based on P2P basis, the second aspect is in our implementation of jSon which uses and adapts P2P network to our needs. The last thing which can increase robustness and scalability are data distribution models.

3 Distribution data models

Data distribution models [2] are used for organizing AEs and clients into the schema which specifies the ways of transfer of data among AEs and users in the concrete collaborative session. The robustness and the scalability of the collaborative environment is basically given by these models.

4 Peer–to–Peer Networks and JXTA

We chose P2P network from many types of the overlay networks for an implementation of the overlay layer of the AE network. P2P networks are providing to us their own identification mechanism. They are defining message routing mechanism. They have mechanism to solve failure of nodes. Briefly, P2P network is a virtual network on the top of the underlying network. Peers are only exchanging management, notification and query messages by the P2P network, concrete content is transferred directly between peers using only underlying network.

P2P networks can be divided into the several categories. We can divide them by their topology or by searching mechanism which they are using. Each category has its own advantages and disadvantages especially in case of robustness and scalability. We have chosen P2P with hybrid type of topology because this P2P networks have balanced level of scalability and robustness. They also allow searching for known (hash search) and unknown content (wildcard search). Hybrid type of P2P needs minimum of management messages to get work.

For our implementation we chose JXTA P2P framework [3]. The JXTA is set of protocols which can create a virtual network overlay on top of the existing physical network infrastructure. The JXTA virtual network allows a peer to exchange messages with any other peers independently of its network location (firewalls, NATs or non-IP networks). Messages are transparently routed, potentially traversing firewalls or NATs, and using different transport/transfer protocols (TCP/IP, HTTP) to reach the receiving peers. The JXTA network allows peers to communicate without needing to understand or manage complex

and changing physical network topologies allowing mobile peers to move transparently from on location to another. The JXTA virtual network standardizes the manner in which peers discover each other, self-organize into peer groups, discover peer resources, and communicate with each other.

5 jSon

jSon is a prototype implementation of the overlay network for the network of AEs. As name says it is built under the JXTA framework. It represents two modules of the AE, *Management module* (MN) and *Network Information service module* (NIS). jSon does not represent these modules exactly. It is a stand alone modular application which can be plugged into the AE. jSon also implements client side of AE network. On fig. 1 we can see example of AE network using jSon.

At the end of jSon designing we had a list of ideas about functionality of the network of AEs. Our ideas were related to building a collaborative environment with synchronous video and audio transmissions.

1. All AEs and all clients belong to one group.
2. When the conference is created, there is also a subgroup of initial group created which will contain clients and AEs participating on this conference. Each conference is described by its topic and by its content description.
3. When the subgroup is created it contains only one AE and one client. AE publishes information about this particular video conference into the initial group and also scans other AEs and tries to invite them to this subgroup so that they can stand as backup AEs.
4. Clients can find this conference in AE network and can join to the subgroup. They can also bring their actually known AEs to this subgroup. Every invitation of AE depends on the configuration of each client and AE. Some AEs can have strict rules such as they are only a backup AEs.
5. Each AE participating on the conference represents one element in computation of data distribution model.
6. During the video conference AEs can measure distance among AEs and clients (commonly evaluated by RTT). On the basis of measurement AEs can initiate reorganizing of the data distribution model.
7. When some failure occurs on the network or some AE fails in the running video conference, other AEs will recognize it and initiate reorganizing of the data distribution model. This reorganization has to be made in range of seconds.
8. Each AE has information about itself. This information for example contains maximum number of users that can be served, maximum available bandwidth and available content which can be provided.

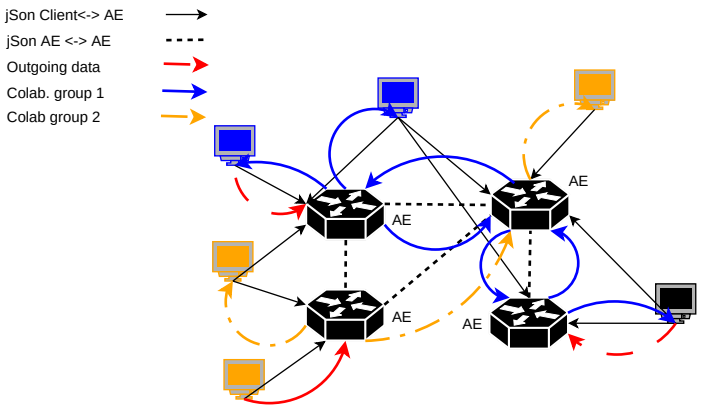

Fig. 1. Example of jSon network architecture

6 Implementation of jSon

jSon is implemented by using C implementation of JXTA because whole AE is written in C language. We had to do some changes in the JXTA framework to better fit our needs. Especially we removed some bugs and added special functionality, like getting bootstrap peers from the web servers. We also changed default parameters of JXTA to be more flexible (quicker reaction on failures occurring in the AE network). jSon is modular like AE, it consists of following modules:

- *ID generating* – Each entity in JXTA network has its own identity called ID. This module can override default behavior of ID generating in JXTA. In the future we would like to generate IDs on the basis of locality. There are two solutions how to handle the locality problem. First of all we can follow pure DHT networks searching mechanism and generate IDs for closer entities (in the sense of physical distance) with similar prefix as long as possible. The second solution is to have some naming service and adapt ID generating to this service.
- *Information center* – Information about available content and capabilities of AE are managed by this module. This module gathers information about host machine, operating system, capabilities of AE, etc. It also holds information about available content which means currently running collaborative sessions and capability of processing of data which goes through the AE. This module is also responsible for publishing information about itself into the AE network.
- *Messenger center* – This module manages messages handling. The module consists of two parts. The first one is used for sending messages and the second one is used for receiving messages. The messages can be sent to all

peers within one group or to particular peer or set of peers. The messages have their own special format where each message has its own name, priority and parameters. Priority determines order of processing of messages received in the same time on the receiving side. Priority also determines the type of message. If the priority is under one thousand then the message is control message, if the message priority is greater then one thousand, it is an informational message. The message is parsed by this module and data are sent to the proper function which processed message on the receiving side.
- *Monitor center* – This module is used for gathering of information neccessarry for JXTANetMap [4] application. JXTANetMap is a tool for visualization of JXTA network. We had special requirements on the monitoring that is why we made some changes to the JXTANetMap.
- *Group management* – It manages peer groups. The groups are founded dynamically on demand, for example group of AEs that supports some transcoding. Each peer group is described by structure which contains all necessary information to identify the group, to get the purpose of the group and to get participating AEs.
- *Discovery* – This module handles all discovery service messages. It can react on each incoming message. When the peer is searching for some unknown information (for example wildcard search) in the network it sends a discovery message. This message is propagated within peer group and every peer can react on this message using this module. This module also handles discovering of peers in the network. Group management module uses this module for updating list of peers.
- *Diagnostic* – This module is used just for debugging and testing purposes. It contains a set of functions testing message sending, getting information about peer status, about connected peers and about actual state of peer view.

7 Tests

We made some tests to see how AE network will work in "near" real operation with jSon prototype. The first set of test was focused on scalability of AE network. Fig. 2 shows influence of growing number of peers on the number of management messages in the whole network. The red lines indicate the time when new jSon peer entered the network. The graph shows that the number of messages grows nearly linearly as a number of peers in the network.

Fig. 3 shows number of messages sent and received by one concrete peer. The data for this graph were fetched from the previous test. This means that a peer in the previous test received and sent constant number of management messages while the number of peers in the network grew.

We established AE network which consists of eight jSon peers and let it idle for about forty hours. The graph 4 shows number of messages exchanged between jSon peers in minutes intervals. We can see that after initial messages exchange the number of management messages stabilized.

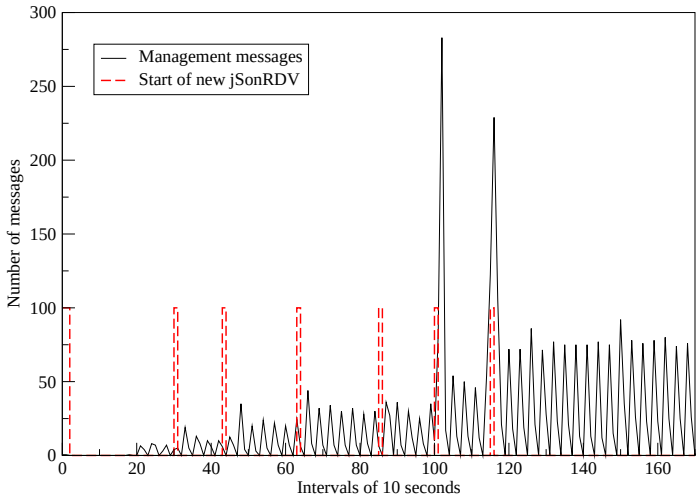

Fig. 2. Influence of growing number of peers in the network on a number of management ı

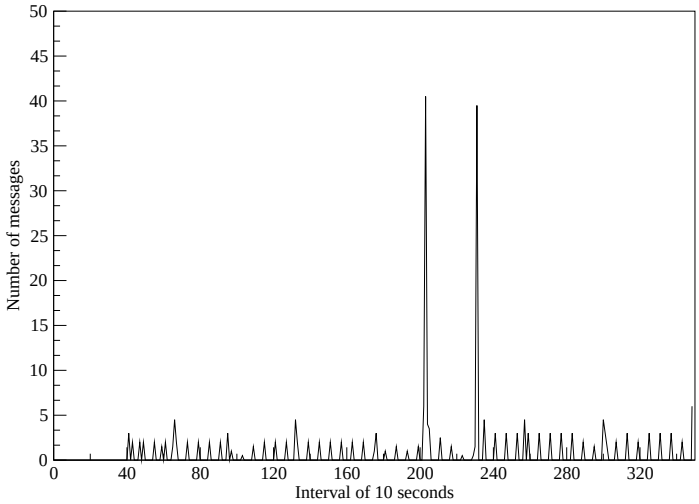

Fig. 3. Number of messages sent by one peer

Another test was focused on the robustness. Eight jSon peers were started and let to establish the network. Randomly two peers were chosen, the first peer was sending every second message to the second peer (sending of messages is represented by black line in the graph). The green line in the graph 5 represents starting of second peer, the red line represents stoping of second peer. As we can

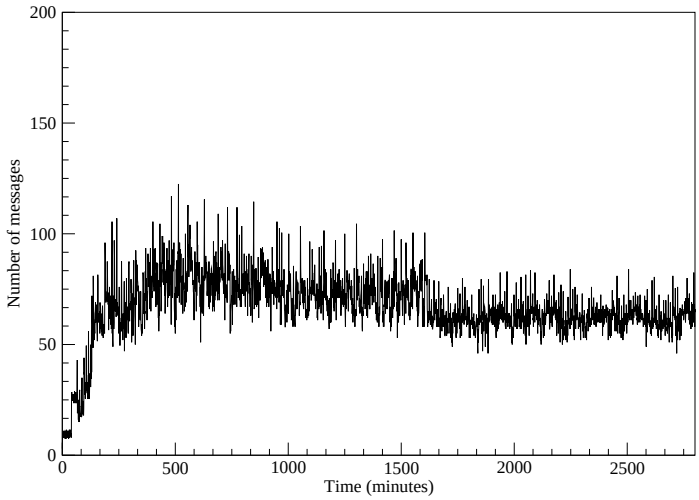

Fig. 4. Long term measure of number of management messages in the AE network

see when the second peer disconnected and after some time it connected again it started to receive messages immediately (under one second). The high black peaks means that the undelivered messages to the second peer were delivered to hi

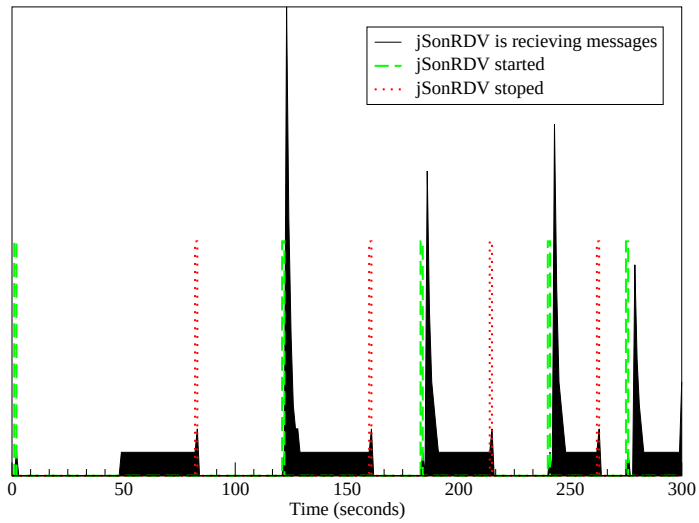

Fig. 5. jSon robustness test

We also measured performance of the JXTA comparing RTT (Round Trip Time) of standard ping and JXTA ping. There was an average difference about 100 milliseconds. This value is to small that is why we do not worry about performance of JXTA and jSon.

8 Future work

Our main goal is to finish implementation of jSon. We hope that JXTA-C will be also finished soon. Another area were we would like to do some work is security, especially secure joining to the infrastructure and secure message transport. Generally we would like to implement AAA (Authentication, Authorization, Auditing) into the jSon. We also aneed some GUI (Graphic User Interface) application on the client side.

9 Conclusions

Distributed system for multimedia data distribution based on AEs using P2P technologies showed that this solution has a good future. We think that the equilibrium among robustness, scalability and efficiency in overlay networks was achieved by selecting JXTA framework. JXTA is very promising among P2P technologies.

In this paper we described what AE is and we proposed the design of AE network. We briefly discussed P2P networks and JXTA P2P network. We presented the prototype of jSon system which is built on the top of the JXTA and it is used for network of AEs enhancement. We discussed implementation of jSon and showed tests made with jSon. Tests showed us that this prototype has a good preconditions to be reference implementation of AE network.

Because jSon implementation was accompanied by serious issues with C implementation of JXTA we could not finish jSon implementation and all tests that we would like to do.

References

1. Hladká, E., Holub, P., Denemark, J.: User empowered programmable network support for collaborative environment. In: ECUMN'04. Volume 3262/2004 of Lecture Notes in Computer Science., Springer-Verlag Heidelberg (2004) 367 – 376
2. Holub, P.: Network and Grid Support for Multimedia Processing. PhD thesis, Masaryk University in Brno, Czech Republic (2005)
3. Traversat, B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J.C., Pouyoul, E., Yeager, B.: Project jxta 2.0 super-peer virtual network (2003) http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf.
4. Abdelaziz, M., Hugly, J.C., Bryson, D., Jan, M.: Project jxta netmap (2005) http://jxtanetmap.jxta.org.