

Horizontal Address Ontology in Internet Architecture

João Henrique de Souza Pereira, Sergio T. Kofuji

Dept. de Eng. de Sistemas Eletrônicos (PSI)
USP – University of São Paulo
São Paulo-SP, Brazil
joahs@usp.br, kofuji@pad.lsi.usp.br

Pedro Frosi Rosa

Faculdade de Computação (FACOM)
UFU – Federal University of Uberlândia
Uberlândia-MG, Brazil
frosi@facom.ufu.br

Abstract—The ontology is used in different areas of computer systems for the explicit representation of conceptualizations, but there are few studies on the application of ontology in the intermediate and lower layers of computer networks. This paper extends the studies on ontology for the network and transport layers of the TCP/IP architecture, to address the human needs that demands technological requirements for the distributed systems. These systems have to address hosts, users and applications. This work presents one proposal to the horizontal addressing in the Internet architecture to collaborate in the improvement of distributed systems.

Keywords—Computer Networks; New Internet; Post IP; TCP/IP Architecture

I. INTRODUCTION

Applications in distributed systems in general are created to meet the human needs, for example the communication needs. These human needs in the real world generate demands for the technological systems and the explicit representation of this conceptualization, in the computer systems, belong to the ontology studies area [1].

In the computational system, the ontology is used in areas such as data base, artificial intelligence, software engineering, and information systems, among others [2]. In the distributed systems area normally it is used in application layer, and not for the other layers of networks.

The other layers, however, are components of the distributed systems and so, improvements in these layers can contribute in the development of the networks. This possibility of contribution in the distributed systems and consequently better meet the human needs, motivates this research, and the target is to collaborate for the new generation Internet and post IP by the use of ontology for intermediary layers in the Internet architecture, as a solution for horizontal address in the networks with the intention to reduce the protocol complexities and neighborhood dependencies that lies in the hierarchy used by the IP.

This paper is organized as the following: section 2 shows the relationship between the applications and the technological requirements they demand; section 3 shows the ontological problems of the protocols used in distributed communication; section 4 proposes alternatives for horizontal address in the

new generation Internet; section 5 shows final considerations over this paper and indicates suggestions for future work in this research area.

II. NEEDS AND REQUIREMENTS OF DISTRIBUTED APPLICATIONS

The applications are developed to support the existing demands in the real world, and for so they demand various technological requirements. These requirements may vary according to the needs and can change along the time, as well as the needs may also change. Therefore, the needs modify directly the technological requirements, which will then modify the construction and complexity of the computer systems.

In a general way, a functionality requested by an application has to have a group of requirements to fulfill it. The applications for communication, in general, are computer networks' area responsibility and its association with the technological requirements takes place in the instant T, according to Fig. 1, in which the application A can be requested to meet a group of needs N that consequently to be fulfilled will have R technological requirements.

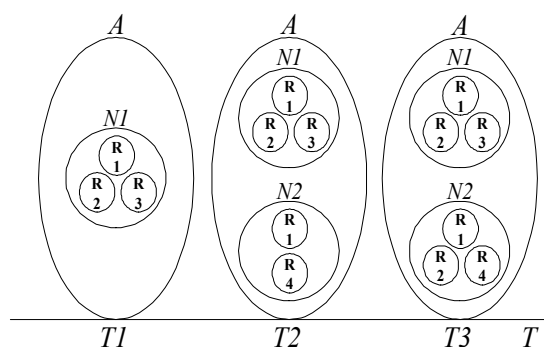


Figure 1. Association between needs and requirements of applications.

There are many communication needs and they yield a series of distinct technological requirements that have to be supported by the networks, which therefore consequently results in technological complexities of different degrees. The technological evolution seeks to better satisfy these needs, or even make happen requirements not yet supported. One example of a need that still has to be improved is the multicast,

since there are difficulties for solutions to support it in the distributed systems. The application needs are related to technological requirements which they demand and this relationship can be represented in a formal way in distributed systems. It is not the purpose of this paper to detail such formal representation. However, and taking its importance, a little approach of this representation will be given in the next subsection.

A. Relationship between Application Needs and Technological Requirements

In a general way, a need requested by an application has a set of requirements, which determines the establishment of a formal relationship between them. For example, for a VoIP (Voice over IP) application to meet the communication need for audio, in real time, some requirements are: real time, low jitter, audio sorting, and duplex communication.

We can notice that such requirements can change in different moments in time, by the modification of needs in different situations. For example, in a teleconference a half-duplex communication can be required, in a moment t_2 , to avoid audio return interference, or in a private conversation, the cryptography can be requested in a moment t_3 . For this example, a possible relationship formalization, according to axiomatic representation defined by [3], is:

$$\text{VoIP}t_1 \rightarrow R \wedge \neg J \wedge A \wedge D$$

$$\text{VoIP}t_2 \rightarrow R \wedge \neg J \wedge A \wedge H$$

$$\text{VoIP}t_3 \rightarrow R \wedge \neg J \wedge A \wedge D \wedge C$$

R - Real Time

J - Jitter

A - Audio Sorting

D - Duplex

H - Half-Duplex

C - Cryptography

In another way, the same privacy need could be satisfied by a private connection (PC), substituting the cryptography requirement, what would lead to a change in the axiom t_3 to:

$$\text{VoIP}t_3 \rightarrow R \wedge \neg J \wedge A \wedge D \wedge (C \vee \text{PC})$$

These are not the only requirements necessary for a VoIP communication, but it exemplifies that the change in the needs, in distinct moments, results in changes of the technological requirements necessary to meet the communication by an application. Historically, the technological requirements of communication applications in the TCP/IP architecture are supported, in most part, by IP, TCP, and UDP protocols in layers 3 and 4. Other protocols were also specified for these layers such as RTP [4]-[5] and SCTP [6]-[7], in the transport layer, to increase the performance or as a attempt to solve problems in the distributed communication systems.

III. ONTOLOGICAL PROBLEM OF DISTRIBUTED COMMUNICATION PROTOCOLS

The architectural design of the TCP/IP solved splendidly the communication needs addressed to it. On the other hand, the principal protocols of this architecture used for addressing and transport were specified three decades ago, and meanwhile new needs came up, which therefore requested new technological requirements for them. As the expansion of the installed base makes it more difficult to have an architectural

development, the solutions for the new needs were performed with adaptations (workaround) and new protocol specifications in the architecture, what is not always the best solution to solve new requests. This has created gaps in the TCP/IP architecture due to an ontological deficiency (explicit conceptual representation deficiency) for the protocols used for distributed applications, in a way that intermediate layers can satisfy the needs of distributed systems.

A part of the impact in the complexity of protocols of this architecture can be noticed by the analysis of new RFC specification evolution, that consequently reflects in the Internet complexity, since its network elements have to be adequate to the new ones. The view of the quantity of RFC, specified along the years, is shown in Fig. 2. This figure was generated, in this work, by the IETF RFC pages information.

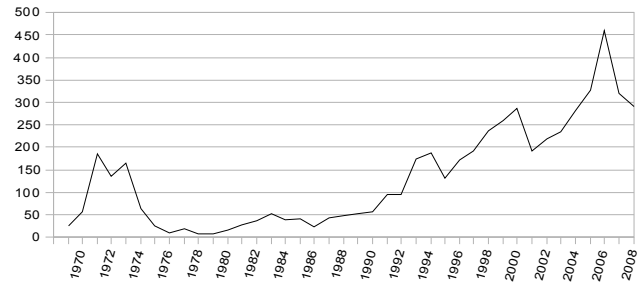


Figure 2. New RFC specified per year.

The analysis of this chart of new specification evolution shows that the mass expansion of Internet architecture use in the decade of 90's, both for companies and people, resulted in more protocols specifications because of the new needs required by the new users. This increase in the new specifications reflects in the rise of complexity of hardware and software of distributed systems and consequently in the Internet complexity architecture as a whole. This can come to a collapse of complexity in a near future, and not just a collapse for the end of addresses in IPv4 to be solved in IPv6, changing the quantity of bit addresses from 32 to 128 [8]-[9].

To contribute to the analysis and development of this architectural problem, this research proposes an approach between the applications and the lower layers, taking into consideration the current technological development and the new needs required by the applications. The biggest benefit of an architectural development is not just in the overhead of the used protocols, but the reduction of complexity and better distributed communication.

The TCP/IP architecture was designed to have dispersed hosts which would still keep the communication even if one of the nodes became inoperative. So, one of the most important motivation in its project was the communication without interruption. Later, with the emerging of new users and new needs, gaps happened in this architecture, and one of the starting points to consider in this discussion is about the address of hosts, applications, and users.

A. Problems in Hosts, Applications, and Users Address

In the TCP/IP architecture, the host's logical address is done by the Internet protocol in the layer 3. For so, it was created an address hierarchy organized by networks and sub-networks using masks for segmentation. For applications, once the host is found, protocols in the transport layer are used, such as TCP, UDP, and SCTP, to find the process by its port.

A quick view over the communication need among people in the real world shows that for some distributed applications the important is localize the user, and not its host or application. For example, in an instant message communication, video conference or VoIP, the user generally wants to communicate with another user and not with an application or host. This could happen when accessing a site with sports content to get information about the score of a soccer game, for example.

This rise in the need for addressing users came with the increase of the Internet use, which became available for people and not just for hosts and private groups. The needs for address among people are still without direct support of protocols from the intermediate and lower layers, and are treated only by the application protocols.

Another gap in distributed application address is in the correct treatment of communication between 1 to many or 1 to all, under the responsibility of multicast and broadcast technologies, respectively. The current applications still have difficulties in some aspects of these technological requirements. For example, in a VoIP conference which happens among 6 people spread over Canada and Brazil, there will be multiple stream flows among the users and according to the VoIP server, there will be multiple flows among it and the users, as in Fig. 3. In the current architecture, there is a difficulty in multicast configuration for this communication.

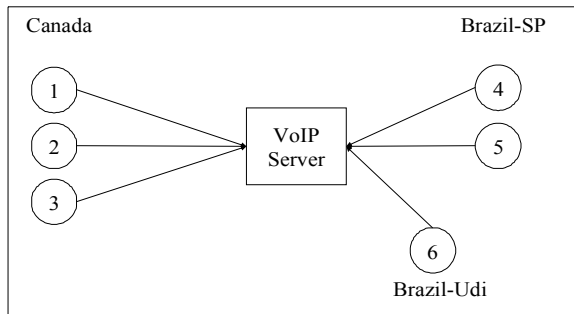


Figure 3. Multiple communication flows among users.

The difficulty to establish this communication in a optimized way in TCP/IP architecture lies in the fact it was not designed for so and also because the support given from new protocol specifications demand a reasonable complexity for its implementation. Therefore, have some solutions limited to specific environments, such as the NGN networks, that use the H.248/MEGACO application protocol to establish VoIP communication by means of a connection that uses contexts and terminations, and so being a context formed by the association of terminations [10].

However, an application using the H.248, when establishing a VoIP context for users' conference establishes flows among them or among the MG (Media Gateway), according to the configuration used for SDP (Session Description Protocol) [11], which results in similar connections to the ones in Fig. 3. One solution that could be more elegant and efficient for addressing would be performed, in a instant t4, with the support of intermediary or lower layers from a request by an application that demanded the multicast (MC) for lower layers.

$$\text{VoIPt4} \rightarrow R \wedge \neg J \wedge A \wedge D \wedge MC$$

This requirement could be requested for establishing a workspace in a way to optimize communication flows among the bigger distances, according to the result found in Fig. 4. A design like this can cause architectural impact in protocols of intermediary layers, as well as in compilations of applications.

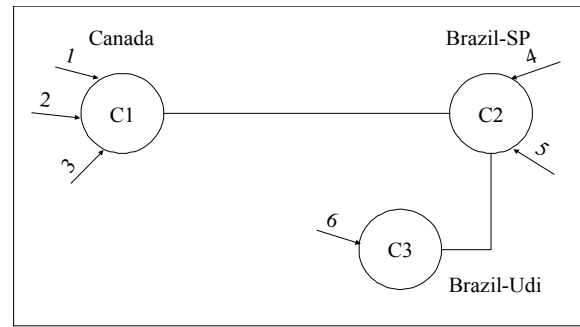


Figure 4. Association between contexts and terminations in a workspace.

This kind of problem found in multicast also happens in the broadcast. It happens due to the IP address hierarchy, which distributes its networks and sub-networks in organized classes according to the constructions of octets of its addresses and masks [12]-[14]. In case the architecture were designed to suit this requirement and address in a horizontal way, such problem could be solved in a structured way and the neighbors would be addressed horizontally, without due to its strong coupling.

This way, the user number 1 could be anywhere in the world, what also could happen with his near neighbor. Such thing does not occur with the IP, because its hierarchy organization in networks and sub-networks, which reflects in the requirement that the neighbors be near, or at least have some connection that enables this (one example is a physical link between two hosts distant or one VPN).

Another gap in this problem that we are approaching refers to the fact that in the TCP/IP architecture, the connection model is done by the transport layer, which for media flow generally uses the UDP which is not oriented to the connection. Even though the SCTP were specified for the transmission of streams, they have not been largely adopted [7]. Even if the media flow used the TCP, which is oriented to the connection, the association of Fig. 4 is not possible for the connection model of TCP, that does not have this support.

The TCP design defines a connection between a client and a server with the use of the 3 way handshake, but it does not define the connection between N clients and one server or

among the N clients for this VoIP application. In the connection between the client and the TCP server, the handshake is done in 3 ways to save the resources use in the server and not to support multiple communication associations, such as the ones necessary in the multicast.

This TCP connection model is shown in Fig. 5. In this figure the server will only hold its resources after receiving the ACK from the client, which will guarantee that the prior packages between client and server were not lost. This way, the client will place the resources first and only after the server receives the client's confirmation it will place them. This mechanism protects the server from placing resources for too many clients whose connections have not been completed successfully [15].

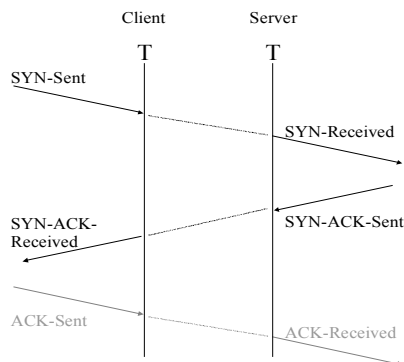


Figure 5. TCP basic 3 way handshake.

An architecture design to address users, applications, and hosts horizontally and with a connection model that allows to satisfy the needs of current applications, such as multicast, can help with the development of distributed systems, in a way to reduce its complexity and enhance its efficiency. The next section presents proposals for this addressing.

IV. PROPOSAL OF HORIZONTAL ADDRESSING FOR INTERNET ARCHITECTURE

An analysis on the real world and its needs of communication is important for an adequate project in the distributed systems area. There is an ever growing tendency for distributed communication among applications, technological devices, humans, and other elements. However, for Internet architecture, there are some complexities in some technological requirements with the use of logical addressing done with IP, due to its hierarchy structure and the TCP/IP architecture itself.

For example, there would be great difficulty for a proposal in which each mobile device and each human had his/her/its own IP, because of the difficulty of controlling human mobility with the hierarchy addressing of IP. Another example of complexity in this architecture is the local networks, which besides the IP, use the MAC address (Media Access Control) to address the network elements (in this case switches and hosts) that make use of physical address in the layer 2.

To the mobility of connections, in the current architecture, it is used mechanisms such as the login, which has similar characteristics of logical addressing, but which is generally

supported by the application layer. This results in the use of 4 distinct addresses controlled by different layers for one single user in the TCP/IP architecture.

1. Physical address (MAC) controlled by the link layer;
2. Logical address controlled by the network layer;
3. Application address (ports) controlled by transport layer;
4. Users address controlled by the application layer.

The existence of this layer overlapping raises the complexity in architecture and makes it more difficult to be optimized in a way to guarantee the communication in distributed systems. Another problem concerning the address overlapping is that among them only the logical one has a hierarchy addressing. The other three (physical, application and user) have horizontal addressing. However, it is the IP address (network layer) which is responsible for addressing hosts in the TCP/IP architecture.

Such overlapping in the use of addresses results in restrictions concerning each one of them. The IP is stuck in its hierarchy structure and restricts the mobility. For example, for a car to communicate while moving, with an IP address, it has to do the handoff to keep the data transfer. This mechanism also occurs in a similar way for handsets with mobile technology, such as GSM (Global System Mobile) or 3G, which in spite of having an associated IP during each data transfer, also has to change the IP in the handoff between distinct networks. Besides that, the handsets also have the address of the device (IMEI – International Mobile Equipment Identification), the address of the SIMCard (IMSI – International Mobile Subscriber Identity), and the user's address.

There are many possible solutions to horizontal addressing in the Internet architecture, but it is important that they guarantee the inter-operability with the current architecture, so not to cause great impact in communication with in use technologies, or this would cause restrictions for its implementation in large scale. The following sub-sections show a summarized description of some possibilities for horizontal addressing in Internet architecture.

A. Horizontal Addressing by Flow Control

The horizontal addressing by flow control can be performed from a structure in which each user, application, or host have an identification. This enables the existence of dynamic structures that meet the needs of applications that came up after the initial TCP/IP architecture design, and it can also facilitate the fulfillment of technological requirements that these needs demand.

For example, is possible to create an identifier to the quantity of characters present in the flow number (or name). Therefore, the size can range from very small to very large, without increasing the network cost by using an address with a fixed size. This way, a flow number can be used to address the users as well as the applications and hosts.

For so, one application, user, or host can request one or more flows and register them dynamically in one cache structure. This way, the network elements responsible for the routing send the data for the destination that has the registered

flow id. In case the network element does not know the flow, it will check the service of names and update its cache table.

The packages with the flow number N are sent to the destination that registered this flow. Also, the holder of a flow N can free it, what will consequently clear the cache tables. In this solution, there can also be the register of flows with fixed identification, and the fixed registration removes from this flow the possibility of dynamic use.

B. Horizontal Addressing by Directory Service

The use of directory service can be an alternative for horizontal addressing in Internet architecture. In this solution, the directory service substitutes the cache structure described previously and allows, besides the horizontal addressing, the control over the others technological requirements necessary for the needs of the applications, hosts, and users. The directory service can be used as complementary way to the solution described above for control of flows. So a network element when updating its flow table also updates the technological requirements that have to be supplied for communication.

For example, a user of an application of instant message communication makes the registration in the directory system and informs that it is necessary the packets delivery guarantee when sending data. Its application recognizes the flow number of the principal service, which can be fixed, and establishes a connection with it through the network elements. So, the application of the destination, checks the directory service and is informed to guarantee the data delivery. When connecting to the destination application, the origin user receives the list of registered users and the information of the ones that are online.

In case the user wants a safe connection, it is possible to register in the directory service the need of data cryptography. This way, the network elements are updated by the directory service and when re-directing packages for the flow that are necessary, they automatically do the cryptography technological requirement. The new network elements, that did the checking to the directory service to localize the flow destination will also be informed about the need to fulfill the security requirement.

This way, through the network structure, the users, applications, and hosts make the registration in the directory service, as in Fig. 6, and the network elements check the directory service, both to localize possible routes of destination, as well as to identify what the communication needs to be fulfilled are.

The network elements in the aforementioned solution have to support the flow identification and, in a similar way to the current switches, keep a table with updated information similar to the ARP (Address Resolution Protocol) table, which is used in the current architecture. In this new table the MAC address information is substituted by the flow identification which has an interface correspondence to which this flow responds to.

This way, having a similar structure to the current ARP table, the mechanisms for addressing in local networks will be easier to be adjusted, as well as additional functionalities such as protection against loop in layer 2 performed by the spanning tree protocol [16]-[17].

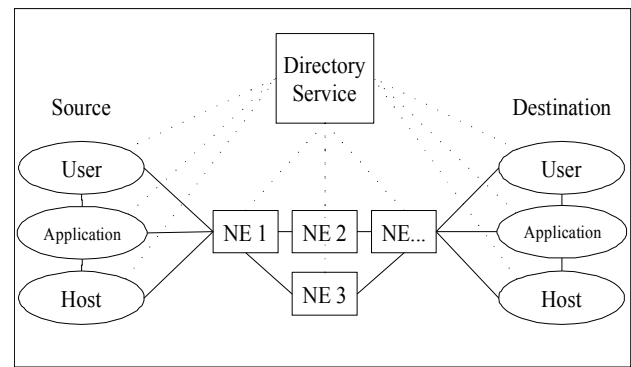


Figure 6. Horizontal Addressing by Directory Service.

For the addressing, besides the local network (in the world web), the directory service informs which of the network elements respond to one flow and also what the average respond time is for each network element. This way, it is possible to identify the best route for future packages. This mechanism can also be used in local networks, if necessary.

The solution with the use of directory service can allow the horizontal addressing in Internet, in an architectural organized way, allowing the decoupling of existing neighborhoods in the hierarchy addressing. To make this possible it is necessary a formal project for this Internet architecture change with explicit representation of this conceptualization.

C. Horizontal Addressing in Internet without the Use of Client-Server Model

In the TCP/IP architecture the distributed communication is established by the use of the client server model in which a server prepares itself to receive data originated by clients that may be in the same or in different hosts. The TCP also uses the connection model as in Fig. 5 and for the UDP the server prepares itself to receive data without establishing a connection by clients. Even the peer-to-peer communication takes place this way, both for TCP and UDP [18].

The client-server paradigm came up due to the “rendezvous” of two hosts problem, since the computer systems have an order of greatness much larger than humans. Thus, a human trying to establish a communication between two softwares, when initiating the first one would try to localize the second one, whose process still does not exist, what would cause an error message. When initiating the second one, the first one would have already failed, and this synchronism is a problem for the TCP/IP architecture described by Comer in [18]: “The fundamental motivation for the client-server paradigm arises from the problem of rendezvous...”

This problem about synchronism to minimize this “rendezvous” problem in the TCP/IP architecture could be solved in the application layer level, however application developers, and/or the application layer protocols should control the timers, which would result in a significant raise in the development complexity.

The discussion of the client-server paradigm, in an initial analysis, does not have correspondence with the purpose of this

work concerning the horizontal addressing in Internet architecture. However, from the architecture design point of view, it is possible to build distributed applications with communication by horizontal addressing without using the client-server paradigm.

This statement is true according to the conceptual analysis of technological requirements demanded by the distributed applications for the human needs. In this conceptual analysis the technological requirements are for data transfer in a distributed way and not to have a client-server communication. The client-server model is used as a means and not the purpose.

In this case, the purpose in general is the communication, and even though it can go beyond the communication needs to, for example, the mutual collaboration among people, where the communication can be or not a means for such collaboration, which does not change the client-server paradigm condition to continue being a means.

The proposal to solve the “rendezvous” problem in the current architecture without the use of client-server paradigm is a design that substitutes it by a client-client model but not similar to the current peer-to-peer, which uses the client-server model. The client-client model of this proposal is a design for the distributed communication in Internet architecture similar to the human communication system where the speaking and hearing devices are always available (initialized) and the choice to establish a dialogue be interpreted by a superior control.

This way, an application can request the technological requirement of CO (connection oriented), CL, or CL/ACK (connection less with acknowledgement) what keeps the behavior according to the need that originated the data transfer among applications, hosts, or users. Thus, the VoIP communication example of sub-section 2.1, in the t5 instant, can be met by the axiom:

$$\text{VoIPt5} \rightarrow R \wedge \neg J \wedge A \wedge D \wedge (C \vee PC) \wedge CL$$

In this axiom the connection less requirement is demanded to reduce the network cost, however if the quality of service be necessary (for example to fax using VoIP connection by T.38 protocol) the CO or CL/ACK requirements could be used. In this case, in an t6 instant, the change in the need results in the change of connection logic, which can be done by the directory service requesting the change of requirements supplied by the network elements, which can satisfy this request, or not, in case of technological support lack from the requirement demanded.

$$\text{VoIPt6} \rightarrow R \wedge \neg J \wedge A \wedge D \wedge (C \vee PC) \wedge (CO \vee CL/ACK)$$

V. CONCLUSION

The change from hierarchy to horizontal addressing in one new generation Internet architecture can bring benefits in the fulfillment of the needs of the users, applications, and hosts in distributed systems. This change is possible in many ways. However, because it is an architectural change, it presents considerable difficulties in its project and implementation. A project of this nature can be started by an ontological view and

its formal representation to satisfy the needs from the real world in distributed systems.

This paper presents ontological problems in Internet architecture and some possible ways, in this architecture, to change from hierarchy addressing to a horizontal one, with a description of characteristics of each approach presented.

For future work in these studies it is suggested to research other ways of horizontal addressing in the post IP Internet architecture and also a more detailed study concerning solutions with the use of directory service for flow control, or title, and changes in the client-server paradigm. It is also suggested to apply other formal methods for a conceptual representation of the horizontal addressing in one new Internet architecture.

REFERENCES

- [1] T. R. Gruber, “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”, *International Journal of Human and Computer Studies*, 43 (5/6): 907-928, 1995.
- [2] G. Guizzardi, “Ontological Foundations for Structural Conceptual Models”, *Centre for Telematics and Information Technology, PhD Thesis Series No. 05-74, Universal Press, Telematica Instituut, University of Twente*, 2005.
- [3] S. Leśniewski, “Comptes rendus des séances de la Société des Sciences et des Lettres de Varsovie”, *Classe III*, pp. 111-132, 1930.
- [4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, *Audio-Video, Transport Working Group, RFC 1889*, 1996.
- [5] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, *Network Working Group, RFC 3550*, 2003.
- [6] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, “Stream Control Transmission Protocol”, *Network Working Group, RFC 2960*, 2000.
- [7] R. Stewart, Ed. “Stream Control Transmission Protocol”, *Network Working Group, RFC 4960*, 2007.
- [8] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, *Network Working Group, RFC 1883*, 1995.
- [9] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, *Network Working Group, RFC 2460*, 1998.
- [10] C. Groves, M. Pantaleo, T. Anderson, T. Taylor, Eds. “MEGACO Version 1”, *Network Working Group, RFC 3525*, 2003.
- [11] M. Handley, V. Jacobson, C. Perkins, “SDP: Session Description Protocol” *Network Working Group, RFC 4566*, 2006.
- [12] D. Comer, “Internetworking with TCP/IP Vol. I—Principles, Protocols and Architecture”, 3rd ed, New Jersey, Prentice Hall, 1995.
- [13] J. Postel, “DoD Standard IP”, *DARPA Information Processing Techniques Office, USC/Information Sciences Institute, RFC 760*, 1980.
- [14] J. Postel, “DoD Standard IP”, *DARPA Information Processing Techniques Office, USC/Information Sciences Institute, RFC 791*, 1981.
- [15] J. Postel, “DoD Standard TCP”, *DARPA Information Processing Techniques Office, USC/Information Sciences Institute, RFC 793*, 1981.
- [16] M. Seaman, T. Jeffree, Eds. “Media Access Control (MAC) Bridges”, *LAN/MAN Standards Committee of the IEEE Computer Society, ANSI/IEEE Std 802.1D-2004, ISO/IEC 15802-3*, 2004.
- [17] T. Jeffree, Ed. “Media Access Control (MAC) Bridges”, *LAN/MAN Standards Committee of the IEEE Computer Society, ANSI/IEEE Std 802.1D-1998, ISO/IEC 15802-3*, 1998.
- [18] D. Comer, D. L. Stevens, “Internetworking with TCP/IP Vol. III—Client-Server Prog. and Applications”, New Jersey, Prentice Hall, 1997.