2010 Sixth International Conference on Networking and Services

Optimization Proposal for Communication Structure in Local Networks

Eduardo Souza Santos*, Fabíola Souza Fernandes Pereira†, João Henrique de Souza Pereira‡,
Pedro Frosi Rosa† and Sérgio Takeo Kofuji‡

*Department of Electrical Engineering
Federal University of Uberlândia
Uberlândia, MG, Brazil
E-mail: eduardo@mestrado.ufu.br

†Department of Science Computer
Federal University of Uberlândia
Uberlândia, MG, Brazil
E-mail: fabfernandes@comp.ufu.br, pedro@facom.ufu.br

‡Department of Electrical Engineering
University of São Paulo
São Paulo, SP, Brazil
E-mail: joaohs@usp.br, kofuji@pad.lsi.usp.br

***Abstract*—The current TCP/IP architecture has a series of redundancy that reflects on overhead during data transfer, which becomes costly for real time applications. The purpose of this paper is to propose a post IP structure called Fast Integration of Network Layers (FINLAN), which eliminates the use of network and transport layers in networks with layer 2 connectivity. The goal is simplify the way the information is addressed and transferred, optimizing the network structure for distributed systems at the new generation Internet.**

***Keywords*-Network Layers Optimization; Local Networks; Post TCP/IP; Overhead Reduction**

I. Introduction

Applications that need to establish high quality communication with high throughput and low end-to-end delays are good examples of the Ethernet current use, which in this context, it can be noticed that in most of the cases these applications are supported by the TCP/IP architecture. Even with the efficiency of this architecture, it is possible to identify redundancies and obsolete fields in its protocol stack that nowadays carry information that cause unnecessary overhead in the network.

It can be noticed that, for example, the checksum field is used both for the IP and the TCP headers and this could be reduced or even eliminated in certain cases, since the detection and correction of errors is the link layer's responsibility [1]. Besides that, "Type of Service" field could be remodelled, according to current needs, which would allow the network overhead reduction.

In view of such enhancement possibilities in the current TCP/IP architecture, the purpose of this paper is to propose an alternative for this structure, given an architecture that can meet the requirements of current applications in a simplified and optimized way, taking into account the real needs of applications such as VoIP communication, which was first developed about fifteen years after the birth of TCP/IP and, therefore, suffer impacts as loss and delay of packets.

One reason that encourages this initiative is the possibilities to contribute in a field that has very few proposals and whose objective is contribute with the studies in new generation Internet technologies that can hold the applications needs better than the IP, TCP and UDP.

The principal idea of a new structure, called Fast Integration of Network Layers (FINLAN), is eliminate the protocols of network and transport layers which will be possible by re-structuring the link layer (Ethernet) protocol, which will serve directly the application layer.

It is important to emphasize that the proposed remodelling does not have the intention to eliminate the use of TCP/IP protocols, but to make Ethernet packets hybrid using the current structure of layers and the new proposed structure.

This paper is organized in the following way: in Section 2 related works are presented, which served as motivation for the proposed new structure in this article. In Section 3 the FINLAN structure is presented, highlighting its functional features. Finally, in Section 4, a conclusion is presented and suggested future works in this research.

II. Related Work

It is possible to find different communications structures in networks, like ATM and X.25, that were proposed and adopted along the years.

978-0-7695-3969-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ICNS.2010.11

About TCP/IP architecture, generally there are more improvements at the lower and application layers, but there are not so much evolution at the intermediate layers. Among these improvements, it is important mentioning proposals that deal with the deficiencies that were appearing in this architecture, with the advancement of new applications and, consequently, new requirements, like the protocols overhead decreasing [2]. Even so, [3] shows that the recent networks of high speed suffer from overhead protocols, placing them as obstacles for the high performance applications that explore high speed connections, for example in clusters.

In the context of distributed systems evolution, it is worthy mentioning alternative technologies to the Ethernet networks. The Local Area Networks (LAN) of high speed Myrinet is one of them, having less protocol overhead than standard Ethernet networks, supplying more throughput, low latency, and less interference [4].

Another example in the new technologies for high speed networks is the Infiniband. Such technology for high speed interconnection supports new protocols of low latency and high broadband which nowadays only have an inferior performance compared to a Gigabit Ethernet. In [5] it is possible to check the high performance of IP protocol integrated into the IPoIB technology (IP over Infiniband).

Old technologies as X.25 were created with a different layer structure that meets specific requirements as safety and reliability [6]. Another old one is the Frame Relay [7], an evolution from X.25 networks developed to transmit data in a specific architecture, modeled in 3 layers, detached from the TCP/IP architecture. Such structures were proposed some years ago and until today are present in networks.

In this genealogy of technologies, it is necessary to highlight the SNMP over Ethernet specification in the TCP/IP architecture [8]. According to this proposal, the network management protocol can be used over the MAC Ethernet layer, instead of going by the stack of UDP/IP protocols. So the data transfer occurs through a logical mechanism that avoids the need of network and transport layers protocols.

Several studies have been developed, also facing an alternative network architecture: user-level network. The idea is to use techniques that transfer messages directly to the user level, releasing the use of the stack of the operating system and thus reducing network overhead. One example is the techniques of zero-copy [9], used in [10], an architecture of network interface for high speed user level devices and in [11] for communication over InfiniBand.

It is also worthy mentioning proposes in the context of mobile networks, that deal with TCP/IP architecture difficulties, for example TCP congestion windows [12]. Analyzing the network-based mobility management scheme instead of host-based mobility, the work of [13] becomes also an example that the mobility networks need evolutions and changes in their architecture. So, it is possible to realize the proposal about simplify de network layers, shown in this paper, to the context of mobility networks.

Several works have been developed also in the area of next generation Internet with the purpose to propose new address solutions, joined with the search for mobility and safety, according to the works [14]–[17]. In [15], it is presented a new model of inter-connection among network elements through flat routing, and in [16], an architecture is proposed for address which meets challenges such as dynamicy, safety, and multi-homing.

In this context, this paper proposes a post IP study for a structure, so-called FINLAN, which eliminates the use of network and transport layers in networks with connectivity in layer 2, differently from work [17], which proposes the creation of an intermediate identification layer in charge of a new address way. The idea of FINLAN is simplify the way the information is addressed and transmitted, optimizing the network structure and reducing the neighborhoods dependency. This next generation Internet layer structure can help for a horizontal addressing architecture, as proposed in the correlate works discussed in [18], [19].

III. Layer optimization proposal

The creation of an alternative layer structure to computer networks, that can meet the current technological needs, can enables a better use of applications needs, that did not exist yet when the specifications of IP, TCP, and UDP protocols occurred.

A good example is the VoIP applications that was developed in the 90's, around 15 years after TCP and IP protocols came up, and which faces a lot of QoS problems. To solve some of these problems would be necessary to think about the structure of the current networks trying to accomplish adequations to the new technologies.

In this aspect, the optimization of TCP/IP architecture through the redesign of fields that throughout the years have lost their meaning, along with the desire to meet the requirements of new applications is an inherent need in the growing use of communications networks and the appearing of new applications.

This way, the modelling of a new communication structure among applications can be on focus, and thinking about it, an optimized TCP/IP stack is proposed with some changes in the Ethernet layer protocol. Figure 1 shows a comparison between the current protocol stack and the new suggested one. It can be noticed that in the new structure, the packets are delivered directly from the link layer to the application layer, eliminating the transport and network layer protocols.

To realize this change, the initial proposal consists of establishing communication between two applications in distinct hosts enabling the exchange of data with the use of only the information from the interfaces of the networks from these hosts for the addressing, in other words, the addressing of applications in Local Area Networks is done with

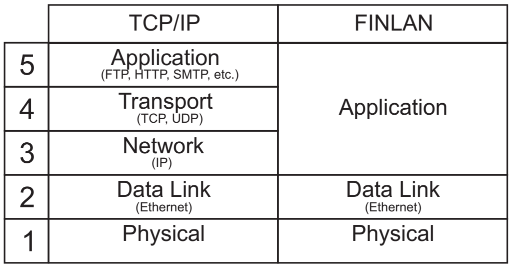

Figure 1. Comparison among the protocol stack

the physical addresses from the machines (MAC Address), direct to the processes without the TCP or UDP ports.

So, to meet the requirements of this new structure, it is proposed some changes in the Ethernet heading in a way that it can still support the TCP/IP structure and also allow the use of the new layer structure that will be named FINLAN. Thus, the separation among packets that use the TCP/IP layer structure and FINLAN will be performed through EtherType field from Ethernet heading. Hence, the transfer of packets that have the designated EtherType for the new layer structure will use this new communication way, based in the direct addressing of applications through communication flows in networks with connectivity in the layer 2.

This proposal was performed in a way that the current structure of Ethernet heading is kept, with fields that consist of identification of the number flow bytes, the packet, and the sequence number, and the fields responsible for transporting these values, thus the heading of a FINLAN application can be described by the Figure 2.

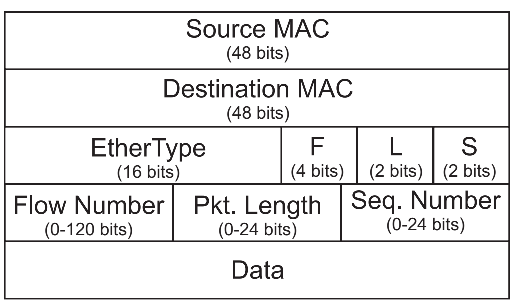

Figure 2. Ethernet heading structure for FINLAN applications

The identification bits contain three fields, “F”, “L”, and “S”, which are the number of bytes used in the fields “Flow Number”, “Packet Length”, and “Sequence Number”. The “F” is represented by a nibble, enabling the aforementioned field to have from 1 to 15 bits of size and therefore the field “Flow Number” can have the values shown in Table I.

It is possible to notice that the field “Flow Number” can have values about $2^{120} - 1$, that show the great number of simultaneous connections. Likewise, the “S” and “L” fields inform that the field “Sequence Number” and “Packet

Table I

RANGE OF POSSIBLE VALUES FOR THE FIELD “FLOW NUMBER”

Number of bytes	Range of values
1	0 to 255
2	0 to 65535
3	0 to 16777215
4	0 to 4294967295
...	...
15	0 to $2^{120} - 1$

Length” can have from 1 to 3 bytes of size, in other words, values from 1 to 16777216. And moreover, considering that the “Packet Length” identifies the number of bytes in the data field and in the heading, it is possible to identify a packet which size of 16 Megabytes, that meets the future networks needs.

For the communication between two stations to take place in a network connected in layer 2, a data packet can be addressed using the physical address. However, more than just physical stations, it is also necessary to address the applications.

According to the current TCP/IP architecture, the IP address is used to locate a host in a network and for each IP there is a series of TCP and UDP ports, where different applications run. Thus, an application can be identified by the TCP or UDP port it is using.

According to the proposal of this work, it was developed a way to deal with and manage the communication channel between applications. So it will not be necessary the use of port and IP addresses. Now, the identification of hosts will be done by the MAC address, applications will be identified by a Flow Number, and a Sequence Number identify sessions.

When an application is started, a flow number is associated with it. Such association can be performed in two different ways: in case the application already has a reserved port according to the current architecture, it will have the same flow number. The numbers from 1 up to 65535 (64k) are reserved for correlation with the TCP/IP ports. Otherwise, the application will request that the operational system choose the flow number that therefore will be above 65535.

When one application is initiated, it requests an available flow number from FINLAN daemon from the operational system and it will be in charge of informing the other hosts what flow the mentioned application has. To establish communication with another host, the application needs to create a new thread, which will request from the daemon a sequence number to establish the communication.

This way the thread sends a packet to the application flow number running in the destination host. The destination host, when receiving this packet, will check that there is no communication established before with this sequence number, so it will create a new thread that will use the same sequence number.

After establishing the sequence numbers for the applications in both hosts, the communication can be initiated. A packet sent from host A to host B will have a sequence number related to the application thread that is running. When the packet gets to the destination host, the operational system daemon will deliver the packet to the application that is connected to the specified flow and the thread of this application will receive the data.

The idea is that with this new communication structure, more simplified, will be possible to improve the head with mechanisms like delivery guarantee, security, detection and correction of errors, in a flexible way, according to the needs of each application.

IV. Implementation Progress

In order to validate the proposal of this work, it is necessary the implementation of the suggested structure. So, in a first step, libraries in C language that supply the services and characteristics presented in the FINLAN proposal had been developed in the application layer.

For so, it was used a Linux operation system with Kernel in the 2.6.28-14 version and, how the libraries and applications are at an application level, the GCC (GNU Compiler Collection) 4.3.3 compiler was used.

As the goal is to address hosts without the TCP/IP layers, the library RAW Socket, available in Linux OS [20], was used allowing the directly communication between the application and link layers.

With this application level implementation, it was possible to do comparative tests between FINLAN and the TCP/IP and UDP/IP protocols. These tests are related to the transfer of files with different lengths and were executed in a unique environment. They shown a performance gain related to the overhead decreasing. However, was possible to verify a little packet-loss rate, what reveal the necessity of implementation of a delivery guarantee mechanism for this kind of application.

The use of the RAW socket for this type of communication do not satisfy many security requirements, since the developed library needs that the network adaptor have been changed to a promiscuous mode. Thereby, was decided that is better to implement the FINLAN proposal at a OS level.

Therefore, the actual stage of this work is the implementation of the proposal using the low level libraries of the Linux Kernel. An important point of this level is to become hybrid the sending and receiving of the FINLAN packets, allowing that the source host be able to deal with errors about identification of these packets. If a FINLAN packet cannot be recognized due the lack of the FINLAN stack, the TCP/IP stack will be used, as shown in the Figure 3.

The Figure 3 shows a scheme representing such structure. It is possible to observe that two elements are considered, they will do the selection of the packages according to the protocol in use. The first one, called "Packet Manager",

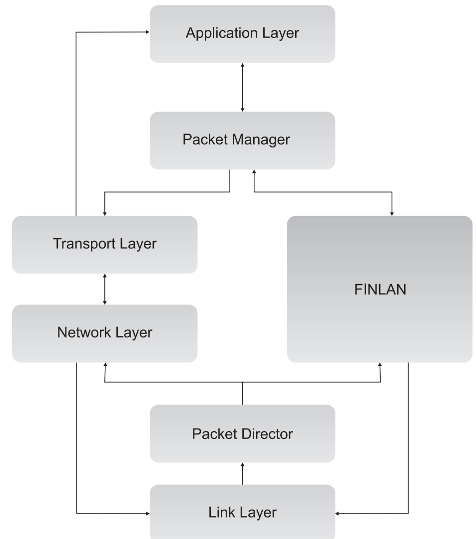

Figure 3. FINLAN model for hybrid communication

is responsible for directing the setting up of the package according with the application layer solicitation, which will inform the transport layer protocol or will inform if the package should be delivered to the FINLAN stack. The other element, named "Packet Director", works when it receives a package and its function is to verify if the package is using the FINLAN structure, in this case such package will be delivered to the FINLAN stack, otherwise it will be sent to the OS standard flow.

In addition, mechanisms like delivery guarantee and error detection, that will be used according to the application needs, are being developed. Thereby, a header of variable length will be used, helping the network overhead decreasing.

At the end of the implementation, is intended to do more conclusive comparative tests, allowing to evaluate the real advantages and disadvantages between FINLAN and the TCP/IP architecture.

V. Conclusions and Future Works

A proposal for communication in LAN networks was presented in this paper for contribution in the next generation Internet studies. It was also shown one way to establish communication between applications through flows that enable an optimized scenario for network use.

In addition, the FINLAN increases the possibility of addressing an enormous quantity of applications and to send very large data packets, keeping the header slim and guaranteeing a good network usage.

Currently, the FINLAN proposal had been implemented at an application level, through C libraries that use RAW socket to establish the communication directly with the link layer.

As future work, it is necessary to design algorithms for guaranteed delivery, security, correction and error detection to FINLAN, according to what the applications need. In parallel to this design, the FINLAN structure has been implemented directly in the kernel of the Linux operational system, to handle the packets in the network interface. The idea is that FINLAN stack will be implemented in a hybrid way, allowing to redirect both IP and FINLAN packets to their respective stack. This will allow the FINLAN approach interact with the existing one.

In this kernel implementation is necessary to do performance tests in real environments with different types of packets and network collapses, providing a comparative data between FINLAN and the architecture TCP/IP in local area networks.

REFERENCES

[1] A. S. Tanembaum, *Computer Networks*, P. Hall, Ed., 2002.

[2] D. Clark, V. Jacobson, J. Romkey, and H. Salwen, "An analysis of tcp processing overhead." *IEEE Commun Mag*, 1989.

[3] H. Jin and C. Yoo, "Impact of protocol overheads on network throughput over high-speed interconnects: measurement, analysis, and improvement," *J. Supercomput*, vol. 41, pp. 17–40, 2007.

[4] A. Barak, I. Gilderman, and I. Metrik, "Performance of the communication layers of tcp/ip with the myrinet gigabit lan," *Comput Commun*, vol. 22, pp. 989–997, 1999.

[5] R. E. Grant, M. J. Rashti, and A. Afsahi, "An analysis of qos provisioning for sockets direct protocol vs. ipoib over modern infiniband networks," *Proceedings of the 2008 international Conference on Parallel Processing*, 2008.

[6] A. Malis, D. Robinson, and R. Ullmann, "Multiprotocol interconnect on x.25 and isdn in the packet mode," *RFC 1356, BBN Communications, Computervision Systems Integration, Process Software Corporation*, 1992.

[7] T. Bradley, C. Brown, and A. Malis, "Multiprotocol interconnect over frame relay," *RFC 1490*, 1993.

[8] M. Schoffstall, C. Davin, M. Fedor, and J. Case, "Snmp over ethernet," RFC 1089, Rensselaer Polytechnic Institute, MIT Laboratory for Computer Science, NYSERNet, Inc., University of Tennessee at Knoxville, Tech. Rep., 1989.

[9] H. Kitamura, K. Taniguichi, H. Skamoto, and T. Nishida, "A new os architecture for high performance communication over atm networks - zero-copy architecture," *Proceedings of the 5th international Workshop on Network and Operating System Support For Digital Audio and Video*, 1995.

[10] T. von Eicken, A. Basu, V. Buch, and W. Vogels, "U-net: a user-level network interface for parallel and distributed computing," *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pp. 40–53, 1995.

[11] G. Santhanaraman, J. Wu, W. Huang, and D. Panda, "Designing zero-copy message passing interface derived datatype communication over infiniband: Alternative approaches and performance evaluation," *Int. J. High Perform. Comput. Appl.*, pp. 129–142, 2005.

[12] Y. Lu, C. Huang, and T. Sheu, "Three-color marking with mlcn for cross-layer tcp congestion control in multihop mobile ad-hoc networks," *Proccedings of the New Technologies, Mobility and Security 2007 Conference*, pp. 145–157, 2007.

[13] D. Damic, "Introducing l3 network-based mobility management for mobility-unaware ip hosts," *Proccedings of the New Technologies, Mobility and Security 200*, pp. 195–205, 2007.

[14] R. Jain, "Internet 3.0: Ten problems with current internet architecture and solutions for the next generation," *Miliray Communications Conference, 2006, MILCOM 2006*, pp. 1–9, 2006.

[15] R. Pasquini, F. L. Verdi, and M. F. Magalhães, "Towards a landmark-based flat routing," *27th Brazilian Symposium on Computer Networks and Distributed Systems - SBRC 2009, Recife - PE, Brazil*, May 2009.

[16] R. Pasquini, L. Paula, F. Verdi, and M. Magalhães, "Domain identifiers in a next generation internet architecture," *IEEE Wireless Communications & Networking Conference - WCNC 2009, Budapeste*, 2009.

[17] W. Wong, R. Villaca, L. Paula, R. Pasquini, F. L. Verdi, and M. F. Magalhães, "An architecture for mobility support in a next generation internet," *22nd IEEE International Conference on Advanced Information, Networking and Applications - AINA 2008. Okinawa, Japan*, March 2008.

[18] J. H. S. Pereira, S. T. Kofuji, and P. F. Rosa, "Distributed systems ontology," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009, in press.

[19] ——, "Horizontal address ontology in internet architecture," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009, in press.

[20] M. M. Alves, *Sockets Linux*. Brasport, 2008.