Layers Optimization Proposal in a Post-IP Network

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

[‡]*Department of Electrical Engineering*
University of São Paulo, São Paulo, SP, Brazil
Email: joaohs@usp.br, kofuji@pad.lsi.usp.br

[*]*Department of Electrical Engineering*
Federal University of Uberlândia, Uberlândia, MG, Brazil
Email: eduardo@mestrado.ufu.br

[†]*Department of Computer Science*
Federal University of Uberlândia, Uberlândia, MG, Brazil
Email: fabfernandes@comp.ufu.br, pedro@facom.ufu.br

***Abstract*—In this work, a post IP structure is proposed, which eliminates the use of network and transport layers in networks with layer 2 connectivity. The goal is to optimize the network structure for distributed systems at the next generation Internet and to propose a delivery guarantee mechanism to FINLAN packets, that emphasizes the optimized relation between applications and lower network layers. The results compared with Internet protocols are also presented.**

Keywords-Network Layers Optimization; Local Networks; Post TCP/IP; Delivery Guarantee.

I. Introduction

Applications that use the computer networks infrastructure have evolved rapidly in recent years increasing the need to establish communication with high throughput and low end-to-end delays (among other requirements). Many of these applications are supported by TCP/IP (Transmission Control Protocol/Internet Protocol) architecture, which was developed to support the communication when the Internet was used to interconnect a limited number of nodes and the applications, in most cases, were used for simple exchange of messages and file transfers.

It may be noted that, in TCP/IP architecture, there are redundancies and obsolete fields in its protocol stack that increase the network overhead. 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. Also, the Type of Service (ToS) field was remodeled to be used as Differentiated Services Code Point (DSCP).

In view of such enhancement possibilities in the current TCP/IP architecture, the purpose of this work is to propose an alternative for this architecture, given a structure that can meet the requirements of current applications in a simplified and optimized way, taking into account the real needs of applications such as Voice over IP (VoIP) communication, which was developed about fifteen years later than the TCP/IP and, therefore, suffer impacts as jitter and packet delay.

One reason that encourages this initiative is the possibility to collaborate in a field that has very few proposals and whose objective is to contribute with the studies in next generation Internet technologies, that can hold the applications needs better than the IP, TCP and UDP (User Datagram Protocol).

The principal idea of a new structure, called Fast Integration of Network Layers (FINLAN) and introduced in [1], is to 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 this proposal 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.

In this work is also proposing a mechanism to guarantee the data delivery in FINLAN, prefaced in [2]. With this mechanism, the operational system will receive the information over the needs of the applications and guarantee the data delivery, when necessary, without the need to use distinct transport protocols, such as UDP or TCP.

This paper is organized as follows. Section 2 presents the related work and a network ontological overview that motivated this research for a optimized communication structure. In Section 3 the FINLAN structure is presented, highlighting its functional features. In Section 4, the proposal of delivery guarantee to FINLAN packets is detailed. In Section 5, are shown details about implementation progress and in Section 6 the preliminary results by the layers simplification are

discussed. Finally, in Section 7, a conclusion is presented and future works in this research are suggested.

II. Related Work

It is possible to find different communications structures in networks, like ATM (Asynchronous Transfer Mode) and X.25, that were proposed and adopted years ago.

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 deficiencies in this architecture, with the advancement of new applications and, consequently, new requirements, like the protocols overhead optimization [3].

Even so, Jin and Yoo [4] show 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 Network (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 [5].

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 [6] it is possible to check the high performance of IP protocol integrated into the IPoIB (IP over Infiniband) technology.

Old technologies as X.25 were created with a different layer structure that meets specific requirements as safety and reliability [7]. Another old one is the Frame Relay [8], 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 (Simple Network Management Protocol) over Ethernet specification in the TCP/IP architecture [9]. 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.

Also, several studies have been developed 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 are the techniques of zero-copy [10], used in [11], as an architecture of network interface for high speed user level devices and in [12] 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 [13]. Analyzing the network-based mobility management scheme instead of host-based mobility, the work of [14] becomes also an example that the mobility networks need evolution and changes in their architecture.

So, it is possible to realize the proposal about simplified 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 proposal of new address solutions, joined with the search for mobility and safety, according to the works [15]–[18]. In [16], it is presented a new model of inter-connection among network elements through flat routing, and in [17], an architecture is proposed for address, which meets challenges such as dynamicy, safety, and multi-homing.

In this context, this work proposes a post IP study for a structure, called FINLAN, which eliminates the use of network and transport layers in networks with layer 2 connectivity, differently from the work [18], which proposes the creation of an intermediate identification layer for a new address way.

Therefore, 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, as proposed in the correlate works discussed in [19], [20].

A. Network Ontological Overview

The TCP/IP architecture is powerful and flexible to handle different application needs. However, for the last 30 years, the main protocols at the Network and Transport layers have not evolved to support the new application requirements.

This statement is comproved by the evolutionary review of the RFC 760, 761 and 768 described in [19]. In this analysis is verified that the IP, TCP and UDP protocols had not evoluted substantially since 1981. Since the 80's, at the Network and Transport layers, are others specifications, as the IPv6 specified in 1995 and the SCTP (Stream Control Transmission Protocol), in 2000.

These specifications solve some problems in the intermediate layers level, but still remain some gaps (or opportunities) for contributions to improve the Internet communication mechanisms.

Proposals as the horizontal addressing by entity title can contribute to reduce the increasing of the Internet architecture protocols complexity. This complexity is increasing as a result of the new communication requirements that appeared after the specification of the main protocols of the TCP/IP intermediate layers. The Figure 1 built from the Internet Engineering Task Force (IETF) RFC index, shows

the Internet protocols complexity evolution, since the first IETF specification through nowadays [20].

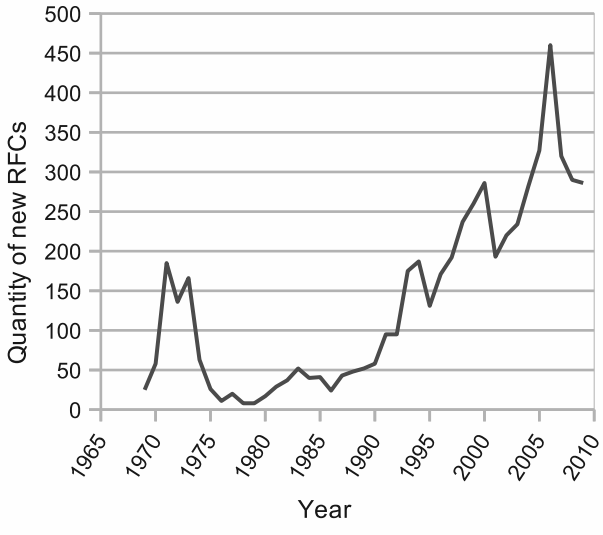

Figure 1. RFCs specified by year

By this graphic information there is a risk of complexity collapse in the Internet because, usually, new specifications demand new technological requirements for some network elements (hardware and/or software). To amplify this discussion, in another direction, the W3C (World Wide Web Consortium) has specified an ontological application architecture for the Semantic Web, and this architecture does not have the service support for semantics in the actual Transport and Network layers.

This ontological architecture for the Semantic Web has the power of the semantic communication limited inside the application layer, as this layer can not send meaning to the Network and Transport layers. The application layer generally just can choose the transport protocol (TCP, UDP, SCTP, etc.), its ports, and set the destination IP.

In fact, using the traditional TCP/IP layers 3 and 4 protocols, the application layer can not inform completely all of its needs as mobility, security, Quality of Service (QoS), Quality of Experience (QoE), and others. In this scenario, the FINLAN, also, is not able to handle some fundamental concepts of ontology, as the "formal representation of one conceptualization" defined by Gruber, neither to understand semantics in a more comprehensible way, as can be done by the use of OWL (Web Ontology Language) Full, OWL DL (Description Logics) or OWL Lite.

At the actual step, the FINLAN studies aims to contribute for a Prove of Concept (PoC) to the application layer be closer to the lower layers. Also, this PoC checks the possibility to the application layer be able to inform the lower layers the necessity of data delivery guarantee, or not. In this way, it is possible to use the same protocol to the communication between different applications, with or without delivery guarantee in a not reliable connection, as the Local Area Networks.

Another FINLAN contribution is to be one step to the possibility to implement the applications addresses horizontally by entity title, to expand the unified addresses to different communication entities like hosts, users, sensor networks, and others, as proposed by Pereira in [21] to a world wide network.

B. Delivery Guarantee Work

Besides the application layer be able to inform the delivery guarantee need, this work also proposes to improve FINLAN with a delivery guarantee mechanism. This can qualify FINLAN to support others next generation Internet requirements [15], in an optimized way.

It is technically complex to guarantee reliable data transmission over the networks. There are different technologies for loss detection and packet re-transmission up to architectures that do not worry about guaranteeing a reliable transmission.

Among the existing technologies it is possible to point out the old Frame Relay [22] as an example of protocol that works in the lower layers and does not worry about guaranteeing the data delivery. The idea is that the application be in charge of dealing with packet loss. The ATM [23] is another example of technology that does not implement the delivery guarantee. In this technology, there is a great trust in the transmission medium.

The ATM architecture is different from the TCP/IP architecture, because in TCP [24] the delivery guarantee can be done in a non reliable transmission medium through packet confirmation. This occurs similarly in SCTP, which is also a transport protocol in TCP/IP architecture.

There is also the MPLS (Multi Protocol Label Switching), which is a low layer protocol with a large capacity of traffic management and therefore, with more reliability, although it is not designed to guarantee that all data packets will get to the destination [25].

Even in the transport layers, it is possible to point out protocols, which do not meet the delivery guarantee requirement, for example, the UDP [26] of TCP/IP architecture. Such fact can be explained by the purpose of each protocol or architecture, that is, they transmit data that do not have a delivery packet guarantee as a necessary requirement.

On the other hand, it is also possible to find solutions that guarantee the data delivery, implemented in different layers. There are also the old X.25 networks [27], which guarantee the delivery based on confirmation of each data packet received.

In more recent technologies, as Myrinet and Infiniband [28], it is also possible to verify a structure, which provides a reliable message transmission through the sending of

messages of destination requiring the missed packets [29], [30].

There are still works in wireless and mobile networks that have solutions to guarantee data delivery due to the flexibility of a mobile host. In [31], for example, a protocol that uses Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) mechanisms is proposed, aiming at low rate retransmission. In [32], there is the analysis of the problem of using variable paths aiming at low loss rate and the proposal for a load balancing algorithm as a solution.

According to the past and current architectures and mechanisms that deal with delivery guarantee or not, this work proposes a flexible approach, which applications can choose if they need or not of this requirement. This possibility to attend the real requirements of applications is the major purpose of FINLAN.

III. Layer Optimization Proposal

The creation of an alternative layer structure to computer networks, that can meet the current technological needs, can enable 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 were developed in the 90's, around 15 years after TCP and IP protocols came up, and 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 communication networks and the future new applications.

This way, the modeling of a new communication structure among applications can be on focus, and thinking about it, an optimized TCP/IP stack is proposed, previously introduced in our work [1], with some changes in the Ethernet layer protocol.

Figure 2 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 network interfaces from these hosts for the addressing, in other words, the addressing of applications in Local Area Networks is done with the physical addresses from the machines (MAC Address), direct to the processes without the TCP or UDP ports.

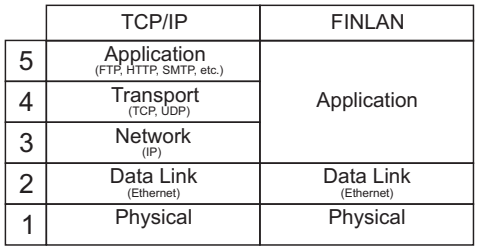

Figure 2. Comparison among the protocol stack

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. 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 on the direct addressing of applications through communication flows in networks with connectivity in 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, the sequence number, and the fields responsible for transporting these values. Therefore, the heading of a FINLAN application can be described by the Figure 3.

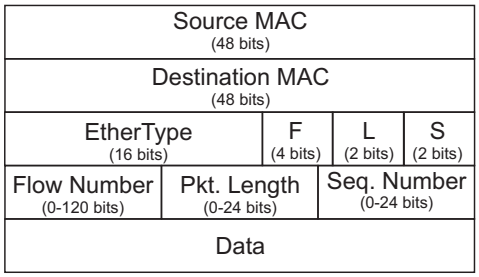

Figure 3. Ethernet heading structure for FINLAN applications

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 bytes 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 Length" can have from 1 to 3 bytes of size, in other words, values from 1 to 16777216.

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 $2^{120} - 1$

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 size of 16 Megabytes, that meets part of 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, 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. In this proposal, the identification of hosts will be done by the MAC address and 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 chooses the flow number that therefore will be above 65535.

When one application is initiated, it requests an available flow number to the FINLAN daemon that communicates with the operational system that 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 receives 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 header with mechanisms like delivery guarantee, security, error detection and correction, in a flexible way, according to the needs of each application. As one example, in the next section is presented a mechanism that realizes delivery guarantee in FINLAN.

IV. DELIVERY GUARANTEE

One of the FINLAN proposes is to meet the application needs. In this context, there is a need to create mechanisms of delivery guarantee, that could be used for services, which require reliable data transmission, such as sending and receiving files.

In this sense, the work shown in [2] proposes a mechanism to delivery guarantee for FINLAN, where the need to make the guarantee is informed by the application via the G flag inserted in the FINLAN header.

Thus, when G=0, there is no delivery guarantee and FINLAN works as described on the initial proposal. When G=1, the field "Packet Number" is enabled with 16 bits. This field is located after the "Sequence Number".

According to Round Trip Time (RTT) algorithm created by Jacobson [33], the delivery guarantee mechanism in FINLAN is done by periodical confirmation according to network behavior characteristic at each instant in time. In this confirmation, the network elements inform the next sequence number to receive the confirmation, indicating the next packet to be confirmed or a packet loss.

This confirmation packet is similar to a keep-alive and does not have data field, having the field L (the quantity of bytes of "Packet Length" field) equal to "00", so the "Packet Length" is suppressed in FINLAN packet and the "Confirmations Quantity" (CQ) field is added, with 8 bits, after the "Packet Number" field.

Depending on the value of the field "Confirmations Quantity", the fields C1, C2, C3, ..., C255 are filled, to inform from 1 up to 255 "Packet Number" not received. Each "Cx" field has 16 bits, since this is the size of the field "Packet Number". For this kind of packet (L=00) the FINLAN structure has the format shown in Figure 4.

Similar to TCP [24], when one keep-alive is sent, a timer is activated and if there is the receive confirmation, the timer is switched off. Otherwise, the keep-alive is re-transmitted. To optimize the use of network in cases of communication failures, when the network element notices that the keep-

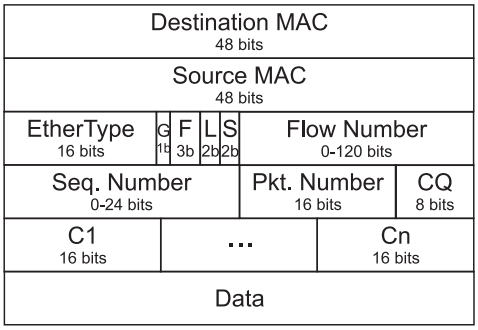

Figure 4. FINLAN confirmation packet [2]

alive is missing, the data transmission is interrupted and will only keep the periodical sending of its keep-alive.

In [2], the timeout interval to re-send a packet is calculated with the use of algorithm created by Jacobson [33], that calculates dynamically the timeout based on continuous evaluation of network performance.

Taking into account the success of the method above, this proposal will be included in implementations that will be performed directly on the Linux kernel, allowing also comparative data in relation to the mechanism implemented in TCP/IP architecture in relation to various scenarios, such as failures on communication network, packet loss and end-to-end delay.

V. Implementation Progress

The FINLAN proposal was implemented in one library using C language and RAW socket. However, the delivery guarantee mechanisms are not in this library yet. Thereby, for the next steps, this library and the delivery guarantee mechanisms will be implemented at a Operational System (OS) Kernel 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 5.

The Figure 5 shows a scheme representing such structure. It is possible to observe that two elements are considered and they will do the selection of the packages according to the protocol in use. The first one, called “Packet Manager”, is responsible for directing the setting up of the package according with the application layer request, 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

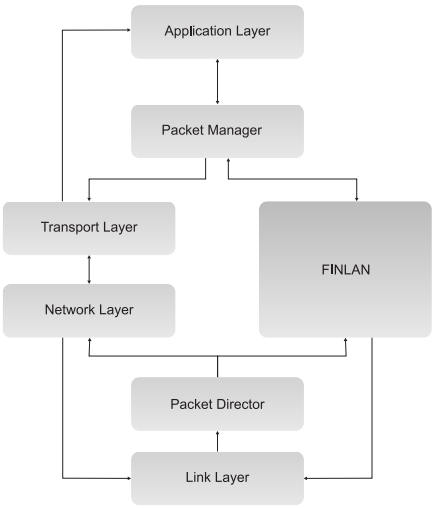

Figure 5. FINLAN model for hybrid communication

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.

The next section presents the implementation results at this stage and some comparative tests between FINLAN and the TCP/IP architecture, at the same environment.

VI. Results

In order to validate the proposal of this work, it was 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 by [34], providing the necessary methods to communicate using FINLAN for the application layer.

For so, it was used a Linux operational system with Kernel 2.6.28-14 and the GCC (GNU Compiler Collection) 4.3.3 compiler was used.

As the goal is to address hosts without the TCP/IP traditional intermediate layers, the library RAW Socket, available in Linux OS [35], was used allowing the directly communication between the application and link layers.

Thus, the library implemented aims to make transparent to the programmer the manipulation of packets that use the FINLAN structure, as well as the creation of RAW Socket. For this purpose, are available to the developer several methods, including:

- *“create_header_eth”*: creates the header for the packet to be sent according to the flow number, packet size, source and destination addresses;

- *"create_socket_finlan"*: responsible for initiating the communication channel using the RAW Socket with the required parameters for sending and receiving FINLAN packets;
- *"send_finlan"*: do the sending of data, being also responsible for the dimension of the packets;
- *"receive_finlan"*: monitors the network interface specified in its parameters. According to the number of established flow and address of the source host, receives the packets and reassembly the file.

It is noteworthy that in the current stage of development, the FINLAN packets are marked with Ethertype 0x0880, which is currently available on the Internet Assigned Numbers Authority (IANA) and the user must inform the MAC address of destination machine.

With this application level implementation, it was possible to do comparative tests between FINLAN and the TCP and UDP protocols. These tests are related to the transfer of files with different lengths and were executed in a unique environment.

Initially, tests were performed aiming to compare sending packets with FINLAN and the TCP protocol, in this case were taken the following values for the sizes of their headers:

- TCP:
  - Ethernet header: 14 Bytes;
  - IP header: 20 Bytes;
  - TCP header: 20 Bytes;
  - Total: 54 Bytes.
- FINLAN:
  - Fixed length (MAC addresses, Ethertype, F, L and S): 15 Bytes;
  - Flow number (equal to port numbers of TCP/IP): 2 Bytes;
  - Packet length (equal to IP packet length): 2 Bytes;
  - Sequence number (maximum capacity): 3 Bytes;
  - Total: 22 Bytes.

It is important to remember that this experiment does not use the full capacity of addressing and packet size of FINLAN proposal, to put these capabilities similar to the protocols of TCP/IP architecture. However, for information that use the full capacity of FINLAN, the header can have the following values:

- Fixed length: 15 Bytes;
- Flow number (maximum capacity): 15 Bytes;
- Packet length (maximum capacity): 3 Bytes;
- Sequence number (maximum capacity): 3 Bytes;
- Total: 36 Bytes.

From the values agreed it was possible to perform tests with files of varying sizes. The Figure 6 shows a graph comparing the total amount of data sent to a file of 10 GBytes, ignoring re-transmissions and packet loss.

Looking to the diagram (Figure 6) may be noted that TCP sends 0.224 GBytes (∼229.9 MBytes) more than FINLAN,

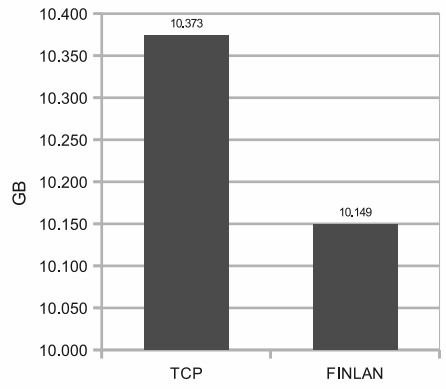

Figure 6. 10GB file transference with FINLAN and TCP

which is equal to about 2% of the total file size. This occurs because the total number of packets sent, which varies due to difference in the size, headers and confirmation.

Despite the occurrence of gain relative to the reduction of overhead compared to TCP, this implementation of FINLAN does not have a mechanism to delivery guarantee, so we did the similar tests using the UDP protocol, since this protocol does not guarantee the package delivery.

Just as in the tests with the TCP, was used standard values for FINLAN to the similar capabilities with UDP, as shown below:

- UDP:
  - Ethernet header: 14 Bytes;
  - IP header: 20 Bytes;
  - UDP header: 8 Bytes;
  - Total: 42 Bytes.
- FINLAN:
  - Fixed length (MAC addresses, Ethertype, F, L and S): 15 Bytes;
  - Flow number (equal to port numbers of TCP/IP): 2 Bytes;
  - Packet length (equal to IP packet length): 2 Bytes;
  - Sequence number (maximum capacity): 3 Bytes;
  - Total: 22 Bytes.

So, as happened in the tests with TCP, the FINLAN provided a gain compared with UDP, relative to the total amount of data sent (Figure 7), reducing the overhead.

Another comparative test with UDP was the percentage of packets successfully received at the destination. In this case, were sent files with a size of 1 KByte to 1 GByte. Performing the sending of each file four times and averaging the rate of packets received, was obtained the graph shown in Figure 8. It may be noted that, as it grows the amount of data and, by consequence, the number of packets sent, the FINLAN structure has a better performance against lost packets.

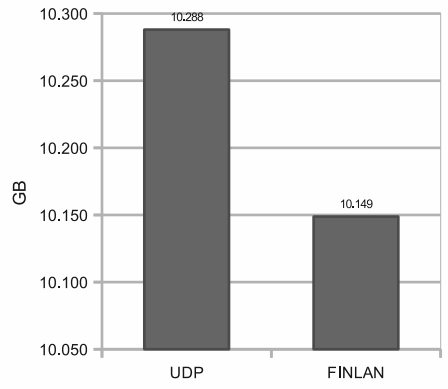

Figure 7. 10GB file transference with FINLAN and UDP

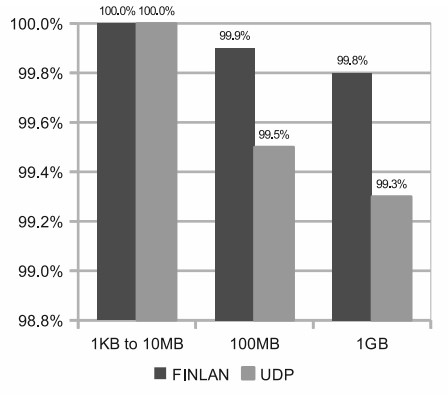

Figure 8. Percentage of packets successfully transferred (UDP vs. FINLAN)

By the tests, FINLAN has good results in comparison with UDP protocol on local networks, related to overhead reduction and lower rate of packet loss. These results brings FINLAN as one possible option for some services, as data streams applications, used in VoIP communication.

VII. 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 enables an optimized scenario for network use and presented the comparative results with the TCP/IP traditional intermediate layers.

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.

This work also contributes with a proposal to guarantee the delivery of packets in FINLAN. By this, the applications do not need to use different transport protocols to have or not data delivery guarantee. In this proposal, the applications only need to inform the operational system their need about data delivery guarantee by using FINLAN library. In turn, FINLAN enables the network overhead reduction by reducing the redundancy and changing the packet confirmation way done by the in use protocols.

So, this proposal is just a step to improve FINLAN with a variety of QoS guarantees, a required feature for technologies for next generation Internet [14]. The idea is append new basic requirements like security and isolation in FINLAN in future works.

Currently, the FINLAN proposal had been implemented in a C library that uses RAW Socket to establish the communication directly with the link layer.

As future work, it is necessary to design algorithms for security, error correction and error detection to FINLAN, according to applications need. In parallel to this design, the FINLAN structure is being 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.

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

References

[1] E. S. Santos, F. S. F. Pereira, J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, “Optimization Proposal for Communication Structure in Local Networks,” *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 18–22, 2010.

[2] F. S. F. Pereira, E. S. Santos, J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, “FINLAN Packet Delivery Proposal in a Next Generation Internet,” *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 32–35, 2010.

[3] D. Clark, V. Jacobson, J. Romkey, and H. Salwen, “An Analysis of TCP Processing Overhead,” *IEEE Commun Mag*, 1989.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] T. Bradley, C. Brown, and A. Malis, "Multiprotocol Interconnect over Frame Relay," *RFC 1490*, 1993.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] J. H. S. Pereira, S. T. Kofuji, and P. F. Rosa, "Distributed Systems Ontology," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009.

[20] ——, "Horizontal Address Ontology in Internet Architecture," *IEEE New Technologies, Mobility and Security Conference - NTMS, Cairo*, 2009.

[21] J. H. S. Pereira, P. F. Rosa, and S. T. Kofuji, "Horizontal Addressing by Title in a Next Generation Internet," *ICNS 2010, The Sixth International Conference on Networking and Services*, pp. 7–11, 2010.

[22] T. Bradley, C. Brown, and A. Malis, "Multiprotocol Interconnect over Frame Relay," *Internet Engineering Task Force Document IETF RFC 1490*, pp. 1–25, 1993.

[23] A. E. Joel, *Asynchronous Transfer Mode Switching*. Institute of Electrical & Electronics Enginee, 1993.

[24] J. Postel, "RFC: 793: DoD Standard Transmission Control Protocol," *Information Sciences Institute of the University of Southern California*, 1980.

[25] E. Rosen, A. Viswanathan, and R. Callon, *Multiprotocol Label Switching Architecture*. RFC Editor, 2001.

[26] J. Postel, "RFC 768: DoD Standard User Datagram Protocol," *Information Sciences Institute of the University of Southern California*, 1980.

[27] "Draft Recommendation X-25," *CCITT Study Group VII*, 1976.

[28] Top500.org, "Interconect Family Share Over Time," retrived 2011-01-10. [Online]. Available: http://www.top500.org/overtime

[29] I. T. Assoc., "InfiniBand Architecture Specification, Volume 1, Release 1.2," 2004, retrived 2011-01-10. [Online]. Available: http://www.infinibandta.org

[30] I. Myricom, "Myricom," retrived 2011-01-10. [Online]. Available: http://www.myri.com

[31] A. Boukerche, D. Ning, and R. B. Araujo, "UARTP - A Unicast-based self-Adaptive Reliable Transmission Protocol for Wireless and Mobile Ad-hoc Networks," *2nd ACM international workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks - WASUN '05, Canada*, pp. 255–257, 2005.

[32] P. Djukic and S. Valaee, "Reliable Packet Transmissions in Multipath Routed Wireless Networks," *IEEE Transactions on Mobile Computing*, 2005.

[33] V. Jacobson, "Congestion Avoidance and Control," *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, USA*, pp. 314–329, 1988.

[34] G. Malva, E. Dias, B. Oliveira, J. H. de Souza Pereira, P. F. Rosa, and S. T. Kofuji, "Implementação do Protocolo FINLAN," *8th International Information and Telecommunication Technologies Symposium*, 2009.

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