***Journal of Advanced Computer Science and Technology**, 1 (1) (2012) 18-31*

www.sciencepubco.com/index.php/JACST

E6 Addressing Scheme and Network Architecture

Dmitry A. Zaitsev, Sergey I. Bolshakov

International Humanitarian University Fontankaya Doroga, 33
65009 Odessa, Ukraine
Email: zsoftua@yahoo.com, 0487983576@ukr.net

Abstract

New E6 addressing scheme for the creation of world-wide networks totally constructed on the base of Ethernet technology is described. Hierarchic E6 addresses with the length of 6 octets are used instead of both Ethernet MAC-addresses and IP-addresses that allows the routing within world-wide networks and cuts overhead of TCP, IP headers; the address space is extended in 16K times regarding IP addresses. Standard Ethernet LLC2 facilities are employed for guaranteed delivery of information. E6 Network Architecture simplifies packets processing algorithms that improves the network performance and QoS.

Keywords: *E6, addressing scheme, network architecture, routing, QoS.*

1 Introduction

Total application of Ethernet technology leads to redundancy of protocols TCP, UDP, IP. But Ethernet MAC-addresses are plain which impedes their usage in world-wide networks. This paper describes E6 Addressing Scheme and Network Architecture which uses uniform E6 address with the length of 6 octets and hierarchic structure [16]. E6 addresses are put into the MAC-addresses fields of Ethernet frame as well as used at the application level. Ethernet LLC facilities are employed for guaranteed delivery of information.

1.1 Historical Notes

In heterogeneous networks TCP/IP protocols played their uniting part based on the mapping of IP-addresses into physical addresses of various technologies. Moreover, while physical networks were unreliable and did not provide facilities of guaranteed delivery, TCP played its central part in the reliable delivery of information over unreliable channels. The cost of this approach is overhead caused by TCP/UDP [1,2] and IP [3] headers, slow algorithms of TCP sliding window, inevitable expense of resources for mapping of addresses (ARP, RARP for IPoverEthernet [4]).

The lack of IP-addresses caused the development of NAT [9] and IPv6 [13] standards. NAT facilities are widely used but lead to expenses caused by additional mapping of IP-addresses. IPv6 is too cumbersome which hampers its practical usage.

IEEE tries to overcome the limitations of scale for Ethernet networks caused by plain MAC-address structure with its new standards IEEE 802.1ah (Provider Backbone Bridge) [18]. The solution is based on duplicate pairs of customer and backbone MAC-addresses and additional mapping of addresses. Such an incremental approach could require triple pairs of MAC-addresses in future.

Recently, Ethernet-adaptors and interfaces of Ethernet switches allow the substitution of the vendor MAC-address by an arbitrary MAC-address which creates conditions for new addressing schemes development.

1.2 Motivation

Ethernet technology becomes a universal networking technology. It dominates in LAN sector. With 10Gbps standards it is widely spread as backbone in campus and metropolitan networks. It replaces STM within backbones of providers with "Ethernet over DWDM" solutions. It ejects xDSL in access networks with "Ethernet for last mile" standards. Moreover, wireless WiFi and WiMAX technologies use the same frame formats.

All the conditions are achieved for the creation of homogeneous world-wide networks completely based on Ethernet technology. And really most of Internet interfaces are Ethernet interfaces at the present time. So the delivery of packets could be done using a pair of MAC-addresses only but each MAC-address should be listed in the routing (switching) table of a device which put the limitation to the network scale.

The lack of IP-addresses is still urgent because IPv6 is not wide spread as a practical solution. Sliding widow algorithms of TCP are too slow for real-time application while fast Ethernet LLC2 facilities [15] are unemployed in the standard encapsulation of IP over Ethernet [4]. VoIP applications generate small packets but their delivery in guaranteed time is complicated by packets overhead with TCP and IP headers.

The IEEE solution of Provider Backbone Bridges (PBB) [18] devised for Ethernet scalability adds new pairs of plain backbone B-MAC addresses to the frame header. It saves the existing IP-MAC mapping and brings additional mappings of B-MAC and C-MAC addresses which delays frames processing on the edge of backbone and expands frames headers.

2 E6-address Structure

E6-address is a uniform network address with the length of 6 octets and hierarchical structure:

```
       0         1         2         3         4
       012345678901234567890123456789012345678901234567
      +-------------------------------------------------+
      |   E6 Network Address   |    E6 Host Address    |
      +-------------------------------------------------+
      \                        /
       ----------\/------------
  Subnet Mask (as the number of bits of E6 network address)
```

E6-address (the same as IP CIDR address [6]) consists of E6 network address (E6NWA) and E6 host address (E6HA). The length of the NWA is variable and given by the Subnet Mask.

The same as IP CIDR addresses E6-addresses allow subnetworking and define a hierarchy of subnets which employ aggregation of E6 host and subnet addresses for the reduction of routers address table size.

The following notation of E6-address is used:

```
x.x.x.x.x.x
```

where x denotes an octet of information.

The Subnet Mask is represented by the number of bits; a slash is used as a separator:

```
x.x.x.x.x.x/m
```

The same as for IP addresses an address with all the bits of E6 Host field equal to zero is considered as E6 Network address and with all the bits equal to unit - as E6 Broadcast address.

Examples:

`1.2.3.4.5.6/40, 10.125.236.17.193.25/36` - host addresses;
`1.2.3.4.5.0/40, 10.125.236.17.192.0/36` - network addresses;
`1.2.3.4.5.255/40, 10.125.236.17.207.255/36` - broadcast addresses.

The length of 6 octets allows the substituting E6-address instead of MAC-address into the corresponding fields of Ethernet frame which brings the hierarchic organization to Ethernet networks. Moreover, the length of 6 octets allows the extension of the address space in 2^14 times in the comparison with IP-addresses (14 because 2 first bits are reserved for broadcast and group addresses by Ethernet standards [14]).

3 E6-addresses Usage

It is proposed to use E6-addresses as a uniform network address:

1) On application level instead of IP-addresses.
2) On Ethernet data-link and physical levels instead of Ethernet MAC-addresses.

Thus, at the source host, the pair of E6 source and destination addresses is passed unchanged from an application via corresponding operating system kernel modules into Ethernet frames which are delivered within the network to the destination host.

3.1 Interfaces of Applications

All the TCP/IP applications can be adopted into E6 networks. The only change is the recompilation of applications with expansion of the address fields from 4 octets (for IP-addresses) to 6 octets (for E6-addresses). The rest of the application interface is saved unchanged.

IP DNS is transformed into E6 DNS with the only difference of 6 octets E6 addresses usage. The system of domain names could be saved unchanged so the transition to E6 networks is imperceptible (transparent) for the customers who use domain names only.

3.2. Interfaces with Ethernet Hardware

E6 addresses are assigned directly to Ethernet interfaces instead of vendor MAC-addresses so at the network periphery usual Ethernet switches can be employed for the delivery of frames. But for the efficient delivery of frames within E6 networks it is proposed the organization of hierarchic E6 subnets and the usage of special E6 switching routers (E6SR). At the first stage, the existing system of IP-addresses can be adopted completely into the last 4 octets of E6 addresses with a special value of the first 2 octets, for instance, 1.0.

4 Stack of E6 Protocols

A summary of E6 stack comparison with OSI-ISO and TCP/IP stacks of protocols follows:

```
+-------------+-----------------------+-----------------------+
|   OSI-ISO   |         TCP/IP        |          E6           |
+-------------+-----------------------+-----------------------+
| Application | HTTP, SMTP, VoIP ...  | HTTP, SMTP, VoIP ...  |
+-------------+-----------+-----------+-----------------------+
| Session     |           |           |                       |
+-------------+    TCP    +-----------+                       |
| Transport   |           |    UDP    |    E6 Concordance     |
+-------------+-----------+-----------+                       |
| Network     |           IP          |                       |
+-------------+-----------------------+-----------------------+
| Data-link   |        Ethernet       |      E6 Ethernet      |
+-------------+-----------------------+-----------------------+
```

The only header which E6 Concordance (E6C) protocol brings into Ethernet frame is the following E6C header:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      QoS      |      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Source Port: 16 bits
    The source port number.
Destination Port: 16 bits
    The destination port number.

In essence, E6C header contains the 4 octets' word of UDP or TCP protocol with the same numbers of ports. For the future development two additional octets containing QoS and TTL parameters are added, which are the reminder of IP [3] header. So, there are 6 octets for E6C header. Header check sums are removed because they are redundant regarding Ethernet FCS field.

4.1 E6 Concordance Protocol

A summary of E6C protocol functions (besides corresponding sockets creation and processing) follows:

- placing source and destination port numbers into E6C header;
- choice of Ethernet LLC2 with Type 1 Operation at UDP call;
- choice of Ethernet LLC2 with Type 2 Operation at TCP call;
- placing source E6 address directly into the source MAC-address field of Ethernet frame;
- placing destination E6 address directly into the destination MAC-address field of Ethernet frame.

For QoS information (usually put into IP header Type of Service field) and TTL transmission, the last 2 octets of E6C header are used.

4.2 E6 Ethernet Driver

E6 Ethernet driver assigns E6 addresses to Ethernet interfaces and employs Ethernet LLC2 facilities [15]:

- Type 1 Operation and UI commands for the delivery of datagrams (datagram mode, functionality of UDP protocol);
- Type 2 Operation and I commands for the guaranteed delivery of data segments (stream mode) with establishing of a data-link connection (functionality of TCP protocol).

For the multiplexing/demultiplexing of E6 frames which are transmitted among frames of other protocols it uses:

1) Special SAP numbers (0xE6) of E6 Ethernet LLC Frame header.
2) Field Type of E6 Ethernet DIX Frame header with the special value 0xE600.

SAP of Ethernet LLC E6 frame 0xE6 and Type of Ethernet DIX E6 frame 0xE600 should be added to the defined types [12].

Format of Standard Ethernet LLC Frame [15]:

```
 6 oct. 6 oct. 2 oct.    1 oct.   1 oct.  1-2 oct.          4 oct.
+------+------+--------+--------+--------+---------+--------+-----+
| DA   | SA   | Length | DSAP   | SSAP   | Control | Data   | FCS |
+------+------+--------+--------+--------+---------+--------+-----+
```

Format of Ethernet LLC E6 Frame:

```
 6 oct. 6 oct. 2 oct.    1 oct.   1 oct.  1-2 oct.          4 oct.
+------+------+--------+--------+--------+---------+--------+-----+
| E6DA | E6SA | Length | E6DSAP | E6SSAP | Control | E6Data | FCS |
+------+------+--------+--------+--------+---------+--------+-----+
```

E6(D/S)SAP = 0xE600;

E6Data - data beginning with E6C header.

It is possible to employ Ethernet DIX Frame for data transmission in datagram mode and in this case 3 extra octets of LLC header are cut. The encapsulation is the same as in the standard IPoverEthernet [4] but with the Type of 0xE600.

Format of Standard Ethernet DIX Frame [14]:

```
  6 oct.   6 oct.   2 oct.             4 oct.
+--------+--------+---------+---------+-----+
| DA     | SA     | Type    | Data    | FCS |
+--------+--------+---------+---------+-----+
```

Format of Ethernet DIX E6 Frame for datagrams delivery:

```
  6 oct.   6 oct.   2 oct.             4 oct.
+--------+--------+---------+---------+-----+
| E6DA   | E6SA   | E6Type  | E6Data  | FCS |
+--------+--------+---------+---------+-----+
```

E6Type = 0xE600;

Note that, supposing each packet contains TCP and IP headers without options, 30-31 octets are cut at LLC E6 Frame usage and 34 octets at DIX E6 Frame usage.

5 Architecture of E6 Network

E6 hosts can be connected to E6 network using standard Ethernet switches. But as far as standard Ethernet switches interpret MAC- address fields as plain addresses it does not bring considerable advantages though the delivery of frames is provided. Such a solution could be used on the periphery of networks only at the beginning stage.

For the delivery of packets within E6 networks usual IP-routers with modified software could be employed. The modification consists in expansion of address field from 4 to 6 octets, abandoning address mapping (ARP, RARP) and the usage of the pair of E6 addresses from Ethernet header for the route solution and forwarding of packets.

But as far as all the interfaces of a router are Ethernet interfaces and are given by their numbers usually named as physical port numbers in Ethernet switches, the device is simplified and looks rather as an Ethernet switch. So it is offered to name devices used for the construction of E6 networks as E6 switching routers (E6SR).

Thus, E6 network is a network constituted by E6SR connected to each other and to E6 hosts. For the delivery of packets (frames) only the pair of E6 addresses is used which is situated into MAC-address fields of Ethernet frames and is unchanged on the whole route of packet's delivery [17]. Note that it is supposed that E6 network is microsegmented so only point-to-point lines are used.

The scheme of E6 packet (frame) delivery follows:

```
+-----------+     +--------+                          +-----------+
| E6 Host X,|     | E6SR1  |          E6SR2...E6SRk   | E6 Host Y,|
| address   |     |        |                          | address   |
| E6X       |     |        |              . . .       | E6Y       |
+-----------+     +--------+                          +-----------+
 |            PortA ^ | PortB              ^  |                  ^
 |                  | |                    |  |                  |
 |  +-----+-----+--+ | |  +-----+-----+--+ |  |  +-----+-----+--+ |
 +->| E6Y | E6X |..|-+ +->| E6Y | E6X |..|-+  +->| E6Y | E6X |..|-+
    +-----+-----+--+      +-----+-----+--+       +-----+-----+--+

                 E6 Ethernet frame (unchanged)
```

Thus, E6SR as an Ethernet switch uses only E6Y for the forwarding of arrived frame and does not change Ethernet header but as an IP-router it interprets the hierarchy of E6 addresses using its address tables which contain subnets given by E6-address and Mask.

6 Architecture of E6 Switching Router (E6SR)

E6SR solves both tasks: switching of frames according to individual addresses of directly attached terminal devices and routing of frames according to E6 network addresses of its routing table. All the interfaces of E6SR are Ethernet interfaces so they are given by the number of physical port only (like for Ethernet switches). Since E6- address is used on all the levels of OSI-ISO model, the task of address mapping with ARP, RARP protocols is annulled. Moreover, E6SR analyses

Ethernet headers only extracting destination E6-address for routing solutions; Ethernet header stays unchanged on all the route of packet (frame) delivery.

E6SR can work in either store-and-forward or cut-through mode. The format of routing table is the following:

```
+-----------------------+-------------+--------+---------+
| Destination E6-address |            |        |         |
+------------+-----------+ Port number | Metric | Options |
| E6-address |    Mask   |            |        |         |
+------------+-----------+-------------+--------+---------+
|    . . .   |   . . .   |    . . .    | . . .  |  . . .  |
+------------+-----------+-------------+--------+---------+
```

It is very convenient to assign own network address directly to E6SR. The corresponding record can be stored into the routing table with a non-existent port number equaling to zero. In this case E6-addresses of directly attached E6 hosts could be assigned automatically at the turning on the terminal device by additional E6DHCP protocol. To distinguish host addresses from network addresses Mask field is used. The value of the Mask equaling to 48 defines an individual address of a host. Little values of the Mask define addresses of E6 networks. Network address 0.0.0.0.0.0.0/0 is used for default route.

So, the mask 47 has no use, the mask 46 defines E6 network with 2 addresses (excluding network and broadcast addresses), the mask 45 - 6 addresses and so on.

For the routing decision two usual rules are applied:

1) Most specific network address (with the longest mask);
2) Smallest metric for the same masks lengths.

E6SR can be employed in the three following ways depending on its connections:

(a) Only individual E6 addresses of directly attached terminal devices - for isolated E6 network with star topology (merely the same as the usual Ethernet switch);

(b) Individual E6 addresses and E6 network addresses - for peripheral networks;

(c) Only E6 network addresses (without directly attached hosts) for backbones.

But mentioned three variants of usage could be implemented by the same device. It is supposed that E6SR routing tables are created either manually or using special additional dynamic routing protocols.

7 An example of E6 Network

Let E6 network has the following structure:

```
+-----+   +--------+
| H1  +---4        |
+-----+   |        |
          | E6SR-1 1-------+
+-----+   |        |       |
| H2  +---3        |   +---5----+   +-----+
+-----+   +---2----+   |        1---+ H5  |
              |        |        |   +-----+
              |        | E6SR-3 |
              |        |        |   +-----+
+-----+   +---4----+   |        2---+ H6  |
| H3  +---3        |   +--4--3--+   +-----+
+-----+   |        |      |  |
          | E6SR-2 1------+  +----- Global
+-----+   |        |                network
| H4  +---2        |                connection
+-----+   +--------+
```

E6 addresses assignment to E6SR and hosts follows:

```
E6SR-1 : 1.2.3.4.5.136/45
E6SR-2 : 1.2.3.4.5.144/45
E6SR-3 : 1.2.3.4.5.152/45

H1 : 1.2.3.4.5.137
H2 : 1.2.3.4.5.138
H3 : 1.2.3.4.5.145
H4 : 1.2.3.4.5.146
H5 : 1.2.3.4.5.153
H6 : 1.2.3.4.5.154
```

Physical port numbers are inscribed at the ends of lines.

Note that, the whole E6 network could be aggregated under

```
1.2.3.4.5.128/42
```

address with subnets

```
1.2.3.4.5.160/45
```

```
1.2.3.4.5.168/45
1.2.3.4.5.176/45
```

left for future development.

The delivery of E6 packets (frames) could be organized using the following E6SR routing tables:

E6SR-1:

Destination E6-address		Port number	Metric
E6-address	Mask		
1.2.3.4.5.136	45	0	0
1.2.3.4.5.137	48	4	0
1.2.3.4.5.138	48	3	0
1.2.3.4.5.144	45	2	1
1.2.3.4.5.152	45	1	1
0.0.0.0.0.0	0	1	2

E6SR-2:

Destination E6-address		Port number	Metric
E6-address	Mask		
1.2.3.4.5.144	45	0	0
1.2.3.4.5.145	48	3	0
1.2.3.4.5.146	48	2	0
1.2.3.4.5.136	45	4	1
1.2.3.4.5.152	45	1	1
0.0.0.0.0.0	0	1	2

E6SR-3:

Destination E6-address		Port number	Metric
E6-address	Mask		
1.2.3.4.5.152	45	0	0
1.2.3.4.5.153	48	1	0

```
| 1.2.3.4.5.154 |   48   |   2    |    0   |
| 1.2.3.4.5.136 |   45   |   5    |    1   |
| 1.2.3.4.5.144 |   45   |   4    |    1   |
| 0.0.0.0.0.0   |    0   |   3    |    1   |
+---------------+--------+--------+--------+
```

Note that the number of hops was chosen as the simplest metric in the above example.

8 Notes on Implementation of E6 Architecture

First of all E6 stack should be implemented within kernels of operating systems: Unix (Linux) [19], Windows etc. The implementation of E6DNS is very advisable.

Then TCP/IP applications should be recompiled regarding new protocol and address family (PF_E6 and AF_E6 respectively) usage. Since all the application interfaces of UDP and TCP (DGRAM and STREAM modes respectively) are saved, the only difference comparing PF_INET and AF_INET is the expansion of address field length from 4 to 6 octets.

As the result E6 applications can work within switched Ethernet in parallel among other protocols. Hosts (and routers) which do not support E6 will drop the corresponding frames. But the full benefits of E6 networking are not reached yet since usual Ethernet switches process E6 addresses in the same way as plain MAC addresses. Simple switched Ethernet can not be expanded into a world-wide network since the plain addresses overflow address tables as far as each individual address should be listed in address table.

Full advantages of E6 networks as well as the possibility of world-wide networks construction are reached at E6SR implementation. The simplest variant of E6SR implementation is the usage of a few Ethernet interfaces within a host and special flag E6_FORWARDING (an analog to IP_FOWARDING) within its operating system. So E6_FORWARDING features should be added to E6 stack.

But the performance of a general purpose operating system could not compete with specialized backbone devices (routers, switches). So, hardware implementation of E6SR is advisable [20].

9 Additional Protocols of E6 Networks

To provide the complete functionality of E6 networks a lot of attendant protocols should be either developed of adopted from TCP/IP or other stacks. First of all the System of E6 domain names E6DNS will make the usage of E6 addresses transparent to the end user. It is offered to recompile TCP/IP DNS [10] regarding expansion of address field from 4 to 6 octets at E6 addresses usage instead of IP addresses.

So, E6 domain name structure will be the same as in TCP/IP. Taking into consideration the necessity of E6-IP gateways development it is offered to employ domain names without suffixes within native network and special suffixes for addressing host of a foreign network. For instance, "ip" suffix to address IP hosts from E6 network and "e6" suffix to address E6 host from IP network:

```
www.mgu.com.ua.ip, www.mgu.com.ua.e6
```

ICMP [5] could be adapted into E6ICMP based on additional SAP and Type numbers, for instance 0xE8 and 0xE800 correspondingly.

Dynamical E6 Host Configuration Protocols (analog to [11]) will serve for automatic assignment of E6 addresses to E6 hosts and E6SR. As far as E6 subnet can be assigned to definite E6SR, E6SR itself could assign E6 addresses to the directly attached hosts.

The most significant for E6 networks development is dynamic routing. Since the passive listening and broadcasting employed within Ethernet lead to unpredictable transitory overloads due to the broadcasting storms, it is offered to adapt dynamic routing protocols of TCP/IP networks. The field of address is expanded to 6 octets and only physical ports numbers are used to identify interfaces. RIP [7],

OSPF [8] and other protocols could be transformed into E6RIP, E6OSPF and so on. E6-IP gateways can work on the base of NAT [9] or proxy-server technology. E6 and IP networks could compete freely using the possibility of information exchange. Security and other kind protocols could be adopted as well to expand the functionality of E6 networks.

Acknowledgements

Thanks also to Kirill Guliaiev, Tatiana Shmeleva, Peter Vorobiyenko, Oleg Nechiporuk, for input contributions used in this paper.

References

[1] J. Postel, *Transmission control protocol*, RFC 793, 1981.
[2] J. Postel, *User Datagram Protocol*, RFC 768, 1980.
[3] J. Postel, *Internet protocol*, RFC 791, 1981.
[4] C. Hornig, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, RFC 894, 1984.

[5] J. Postel, *Internet Control Message Protocol*, RFC 792, September 1981.
[6] Y. Rekhter, T. Li, *An Architecture for IP Address Allocation with CIDR*, RFC 1518, September 1993.
[7] C. Hedric, *Routing Information Protocol*, RFC 1058, 1988.
[8] J. Moy, *OSPF specification*, RFC 1131, October 1989.
[9] K. Egevang, P. Francis, *The IP Network Address Translator (NAT)*, RFC 1631, May 1994.
[10] P. Mockapetris, *Domain Names - Concepts and Facilities*, RFC 1034, USC/Information Sciences Institute, November 1987.
[11] R. Droms, *Dynamic Host Configuration Protocol*, RFC 1531, Bucknell University, October 1993.
[12] *Assigned Numbers: RFC 1700 is Replaced by an On-line Database* / J. Reynolds, Ed., RFC 3232, January 2002.
[13] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 1883, January 1996.
[14] *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access method and Physical Layer Specifications*, LAN/MAN Standards Committee of the IEEE Computer Society, IEEE Std 802.3-2005, Approved 9 June 2005, IEEE-SA Standards Board IP, 417 p.
[15] *Logical Link Control, LAN/MAN Standards Committee of the IEEE Computer Society*, IEEE Std 802.2, 1998 Edition (R2003), 239 p.
[16] P.P. Vorobiyenko, D.A. Zaitsev, O.L. Nechiporuk, “World-wide Network Ethernet?”, *Zviazok (Communications)*, No. 5 (2007), pp. 14-19. In Russ.
[17] K.D. Guliaiev, D.A. Zaitsev, D.A. Litvin, E.V. Radchenko, “Simulating E6 Protocol Networks using CPN Tools”, *Proc. of Int. Conf. on IT Promotion in Asia, Tashkent (Uzbekistan)*, 2008, pp. 203-208.
[18] *Virtual Bridged Local Area Networks, Amendment 6: Provider Backbone Bridges*, IEEE Draft P802.1ah/D4.2, Work in Progress, March 26, 2008.
[19] K.D. Guliaiev, D.A. Zaitsev, “Experimental Implementation of Networking Protocols Stack E6 into OS Linux Kernel”, *Artificial Intelligence*, No. 2 (2009), pp. 105-116. In Russ.
[20] *Report on scientific-research work* "New World-wide Networks Addressing Systems Development", state register number 0108U008900 / D.A. Zaitsev, T.R. Shmeleva, K.D. Guliaiev // Odessa: ONAT (2009), 124 p. In Ukr.