# NETWORK ADDRESSES AUTOCONFIGURATION MECHANISM

Constantin Viorel MARIAN[1], Victor CROITORU[2]

*A mandatory task in telecommunication networks is to configure the addressing plan of all network elements. The paper proposes a possible implementation of a mechanism such that the network equipments/nodes will generate their own interfaces addresses. The addresses calculation is based on unique information related to the network node such the node's individual hardware. Different calculation formulae are provided for each IP class type. The algorithm running on every interface of all network nodes is presented in the paper and also is presented a protocol to be implemented between network nodes to exchange any type of information, here is used for IP network addresses. The protocol is independent of the initial configuration process of each new equipment but helps to put in sequence addresses on a point to point link.*

**Keywords:** IP addresses, automatic calculation, automatic allocation, DHCP, OSI layer 2 protocol, automatic network addresses exchange

## 1. Introduction

As a concept, in any general communication system, it is necessary to uniquely identify all network elements such as core network nodes and ending with interfaces [1, 2]. In majority of cases a human operator has the tasks to design, decide, allocate all addresses and configure manually all elements. This approach has disadvantages: requires high level knowledge, require long planning time, possible conflicts can arise, require long per node configuration time and is error-prone. Another implementation uses a pool to allocate addresses to all network elements. The central server to accomplish this task is found in Ethernet networks and named DHCP (Dynamic Host Configuration Protocol) server based on RFC 1531, 1541 and 2131 [3] for Internet Protocol version 4 and RFC 3315 [4] for Internet Protocol version 6. The DHCP approach eliminates the need of a human operator for addresses management but the addresses are allocated randomly. Also there is difficult to implement it in large networks without broadcast domains.

[1] Ph.D. student, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: constantinvmarian@gmail.com

[2] Professor, Ph.D., Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: croitoru@adcomm.pub.ro

In large networks, any device is uniquely identified, most widely by Internet Protocol address (IP address). Internet Protocol version 4 (IPv4) address is a number of 32 bits and for Internet Protocol version 6 (IPv6) it has 128 bits (RFC 2460) [5]. For Internet Protocol version 4 (IPv4), the addresses are divided in classes ranging from "A" to "E" or reserved (RFC 760 [6] and RFC 791 [7]).

As a very brief summary, the paper is organized as it follows. In 2nd chapter are introduced the conceptual elements of a network node. These elements are the building blocks used to calculate the IP address. In 3rd chapter, using the unique identifiers, are exemplified calculus formulae for different IP class types and a numerical example is given. Chapter 4 presents the algorithm for automatic IP addresses calculation and the protocol for data exchange between nodes. In the 5th chapter, based on the same numerical example, the IP addresses modification is shown due to the protocol interaction. Conclusions are drawn in last chapter.

## 2. Introducing IP addresses calculation using unique information related to fixed elements of a network node

A different approach, presented in [8], to automatically allocate IP addresses, is that the network nodes auto-generate each interface's address (using only information about node's hardware).

Since the addresses generated by one node are independent to those generated by an adjacent node, a communication protocol between any two nodes is required to put the address in sequence.

As advantages, we can list:

- automatic generation of addresses;
- no need for all configurations to be made by human operator;
- no need for address server;
- addresses are not random;
- the addresses on a link between two nodes are in sequence;
- knowing the address of one port,
- the adjacent connected port address can be determined.

A network node is a device having communication interfaces (physical or logical communication port) for interconnection with other network elements. The card is a board with one or more interfaces.

In another paper [8] was introduced a network node definition using the concept of unique identifiers. In Fig. 1, is shown the organization of ports on a card, cards in a shelf, shelves that compose a network node.

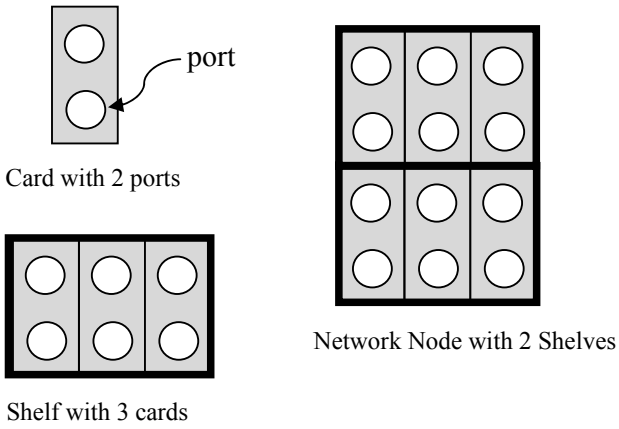


Fig. 1 Example of a network node.

Each communication port can be uniquely identified using identifiers, as shown in Fig. 1, where N is the node identifier; S is the shelf identifier; C is the card identifier; P is the port identifier. Usually a network is organized in one or more areas, each with one or more nodes. The area is identified using A.

In Fig. 2 is presented a network example with directly connected nodes by physical or logical links organized in 3 areas.

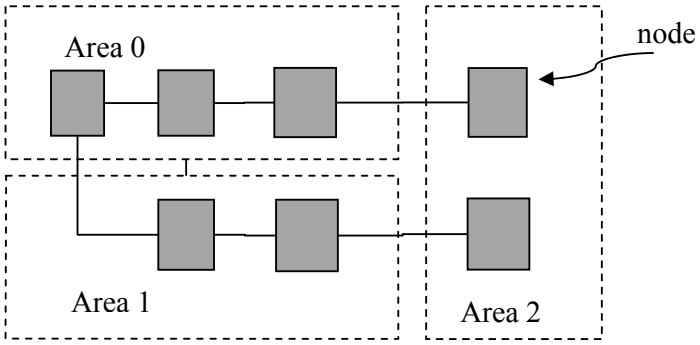


Fig. 2 Network with interconnected nodes (7) organized in multiple areas (3).

When a network is built, the operator will configure the network node with values for:

- A - area identifier,
- N - node identifier,
- Type - IP network.

After that minimal configuration, the process runs automatically by itself. The type of network class is chosen before the nodes are powered according to RFC 791 [7] by the network administrator according to network size.

## 3. IP addresses automatic calculation using unique hardware identifiers and example implementation on a four nodes network - IP addresses values after calculation by each network node

Reference [8] presents a method that enables a network node to allocate itself on each port a unique IP address. This method has four main steps:

Step 1: define of a set of unique identifiers linked to the network device, each identifier assigned to a hierarchical layer in the network (A N S C P);

Step 2: apply a first transformation to these identifiers; the result of the transformation is the calculation of the IP address of the communication port;

Step 3: communicate the addresses between adjacent nodes / devices;

Step 4: for each device, apply a condition to the address received from the adjacent node; if the condition is satisfied, apply a second transformation in order to change one of the existing addresses (so that the addresses are unique and arranged according to the definition of the second transformation).

The above "transformation" is an allocation function for IP addresses and has some mandatory requirements:

- unique ID: the area, and then the node, shelves, cards and ports;
- the combination set of identifiers is unique in whole network;
- area ID number and node ID number are configured by the human operator; the rest of identifiers (shelf, card, port) are read from the network node;
- when the node is started up, it reads the identifier for area and node, and the value of the IP network; the rest of the process is running by itself.

In the next examples, the notation [J] means the integer value of number J.

We define an allocation function (in the next algorithm is presented as "function 1") that uses A, N, S, C, P as unique identifiers and as result calculates a "class A type" IP addresses in the format 10.X.Y.Z:

First octet: 10
Second octet (X): (16 * A) + N
Third octet (Y): (4 * S) + [C/4]
Fourth octet (Z): 64 * (C - (4 * [C/4])) + (4 * P) + 2

where $0 \le A \le 15$; $0 \le N \le 15$; $0 \le S \le 63$; $0 \le C \le 15$; $0 \le P \le 15$.

Since the function is chosen so that is reversible, the unique identifiers A, N, S, C, P are determined from the IP address (10.X.Y.Z) using the following reverse "function2".

A = [X/16]
N = X - [X/16] * 16

S = [Y/4]
C = [Z/64] + 4 * (Y-[Y/4]*4)
P = (Z - [Z/64] * 64 - 2) / 4

As a second example method for automatic allocation of port addresses, for a medium size telecommunication network, a group of multiple "class B type" IP address (or /12 network mask) can be used. Such a network has maximum 16 areas per network, 32 nodes per area, 8 shelves per node, 16 cards per shelf and 4 links per card.

The second method is similar to the first approach except for the function used to determine the exact IP port addresses. The function suitable for the group of "class B type" IP network is shown below, still using the standard format 172.X.Y.Z:

First octet: 172
Second octet (X): 16 + A
Third octet (Y): (8 * N) + S
Fourth octet (Z): (16 * C) + (4 * P) + 2

The identifiers have the values $0 \leq A \leq 15$; $0 \leq N \leq 31$; $0 \leq S \leq 7$; $0 \leq C \leq 15$; $0 \leq P \leq 3$.

A third example method for automatic allocation of port addresses, for a small size telecommunication network, a "class B type" IP address (or /16 network mask) can be used. Such a network supports maximum 8 areas per network, 32 nodes per area, 4 shelves per node, 4 cards per shelf and 4 links per card.

The third method is similar to the first and second approach except for the function used to calculate the port addresses. A suitable function for the "class B network" is presented hereafter, using standard notation 192.168.X.Y:

First octet: 192
Second octet: 168
Third octet (X): (32 * A) + N
Fourth octet (Y): (64 * S) + (16 * C) + (4 * P) + 2

The identifiers have the values $0 \leq A \leq 7$; $0 \leq N \leq 31$; $0 \leq S \leq 3$; $0 \leq C \leq 3$; $0 \leq P \leq 3$.

Another approach, as a fourth example, is the method of automatic allocation of port addresses in a network that conforms to implementations of 31-bit prefixes on IPV4 point-to-point links [9].

This method is useful to any class type of network and has maximum 16 areas per network, 32 nodes per area, 64 shelves per node, 16 cards per shelf and 16 ports per card. In this case, the port addresses are calculated by applying the next function (IP addresses in the format W.X.Y.Z):

First octet: 10
Second octet: (16 * A) + [N/2]

Third octet: 128 * (N - (2 * [N/2])) + (2 * S) + [C/8]
Fourth octet: 32 * (C - (8 * [C/8])) + (2 * P) + 1

This will support a maximum of 512 nodes in a network and up to 16384 network links per node. As an example, in the next network diagram, Fig. 3, is presented an example network with four independent nodes organized in one single area (A=1).

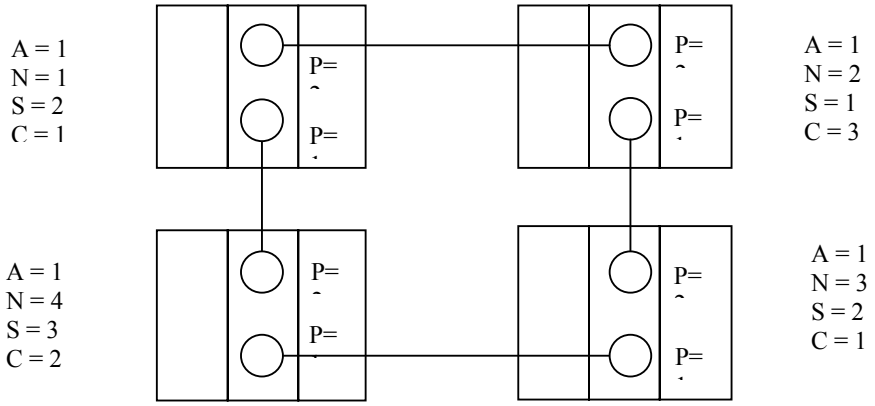


Fig. 3 Network example.

Each node contains one single shelf (S=1 or 2 or 3) populated with a single card (in position C=1 or 2 or 3). Every card has four communication ports. Based on the identifiers for the area, node, shelf, card and port, in Table 1 the default value port IP addresses are calculated according to the "function 1".

*Table 1*

**Calculated IP addresses**

| calculated IP address | A | N | S | C | P |
|---|---|---|---|---|---|
| 10.17.8.78 | 1 | 1 | 2 | 1 | 3 |
| 10.17.8.74 | 1 | 1 | 2 | 1 | 2 |
| 10.17.8.70 | 1 | 1 | 2 | 1 | 1 |
| 10.17.8.66 | 1 | 1 | 2 | 1 | 0 |
| 10.18.4.206 | 1 | 2 | 1 | 3 | 3 |
| 10.18.4.202 | 1 | 2 | 1 | 3 | 2 |
| 10.18.4.198 | 1 | 2 | 1 | 3 | 1 |
| 10.18.4.194 | 1 | 2 | 1 | 3 | 0 |
| 10.19.8.78 | 1 | 3 | 2 | 1 | 3 |
| 10.19.8.76 | 1 | 3 | 2 | 1 | 2 |
| 10.19.8.70 | 1 | 3 | 2 | 1 | 1 |
| 10.19.8.66 | 1 | 3 | 2 | 1 | 0 |
| 10.20.12.142 | 1 | 4 | 3 | 2 | 3 |
| 10.20.12.138 | 1 | 4 | 3 | 2 | 2 |
| 10.20.12.134 | 1 | 4 | 3 | 2 | 1 |
| 10.20.12.130 | 1 | 4 | 3 | 2 | 0 |

If we apply the A, N, S, C, P identifiers to “function 1", we can calculate for each local port its IP addresses (using standard format 10.X.Y.Z).

## 4. Algorithm for automatic IP addresses calculation and protocol for data exchange (such as IP addresses)

In [8] is presented the “automatic allocation of network addresses (AANA) algorithm” which runs on all interfaces of a network node. The AANA algorithm has multiple finite states and is implemented on each node:

1 Start
2 Read A, N, S, C, P
3 Use A, N, S, C, P parameters for “Function 1” to generate port address
4 Send this address to the neighboring node
5 Poll port for received address from the other node
6 Address received from the neighbor? (if YES go to 7, else NO go to 4)
7 Apply logical condition to generated and received address
8 Logical condition true? (if YES go to 9, else NO go to 4)
9 Use “Function 2” (with received address as parameter) to generate new local address and apply it to local port
10 End

As described in [10], the information exchange between adjacent network nodes, defines an OSI layer 2 protocol that is not linked to any higher layer protocols. The protocol is running before any other protocols at upper layer in OSI stack. The protocol exchanges IP addresses between network nodes but can be used for any other type of information. Fig. 4 presents the information exchange between two nodes. In the example below node **j** is recalculating his IP address.

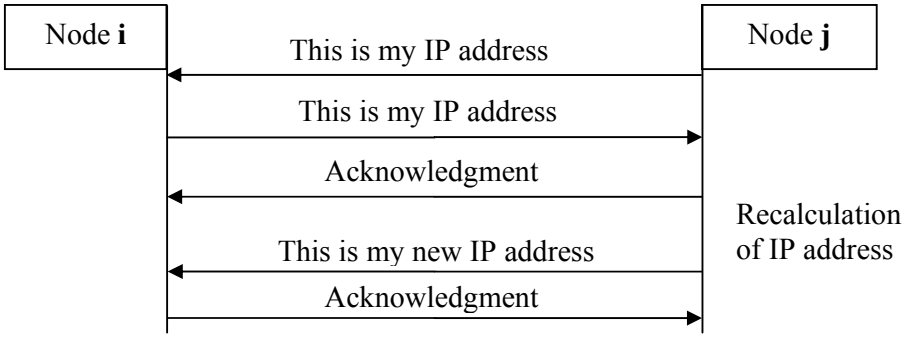


Fig. 4 Information exchange for IP address recalculation (by node j).

The first frame is sent by the originating node towards the connected port on the adjacent node and contains the default (calculated) port IP address. Then

each node polls every port in order to determine if there are new received frames (coming from the adjacent node).

If there are new frames, the node applies a logical condition to make a decision about its default (calculated) port IP address (if the address is re-calculated or not). If condition is true, the originating node uses a function "Function 1" to the received frame that contains the port address. The purpose is to determine a new IP address of the port (the function used is reversible). As an example, the logical condition could be to keep the bigger address and to recalculate the smaller IP address.

Fig. 5 presents the information exchange between any adjacent nodes if in this case the originally calculated port's IP address of node **j** is maintained.

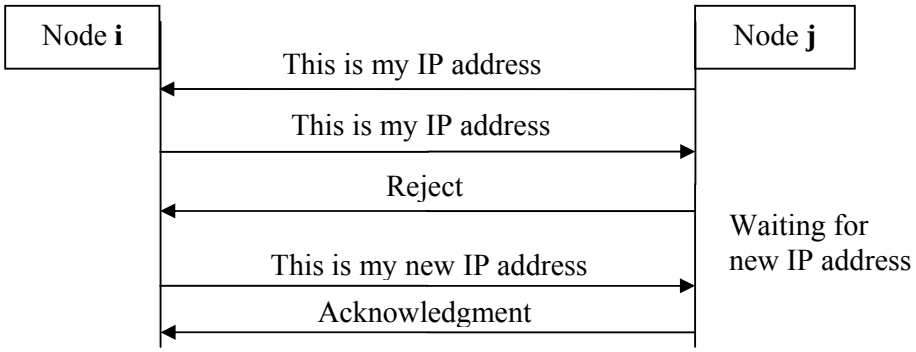


Fig. 5 Information exchange if IP address is maintained (by node j).

The protocol, in order to send and receive information between two endpoints, needs a data field structure. The information is transported between network nodes inside a frame at OSI layer 2 stack. The frame is defined as a multiple structure using type-length-value (TLV) fields:

• Type field (two bytes) indicates if the data value (the last part) is a port address, a node identifier, a version or even another TLV structure.

• Length field (two bytes) indicates the length of next field, the data value.

• Data value field (variable length) - contains the real value. The value is interpreted as the type mentioned in first field. For other purposes, this field could contain not only one but multiple TLV structures concatenated one after the other.

The TLV structure is employed by other communication protocols (such as: COPS, IS-IS, RADIUS, LLDP packets with TLV element used for sending organizational-specific information, RR protocol for GSM networks defined in 3GPP 04.18). This approach is used by some proprietary protocols implemented by vendors such as Cisco Discovery Protocol (CDP), Nortel Discovery Protocol (NDP), Microsoft's Link Layer Topology Discovery (LLTD). A vendor

independent protocol using TLV structures is Link Layer Discovery Protocol (LLDP) IEEE 802.1AB.

## 5. Implementation of a four nodes network - IP addresses values after data exchange between network nodes

In Fig. 3 is presented an example network with communication nodes (numbered from 1 to 4) organized in one single area (A=1). Each node is built using one single shelf (S=1 or 2 or 3) populated with a single card (in position C=1 or 2 or 3). Each card has four active communication ports.

In Table I, based on "function 1" and using the identifiers (area, node, shelf, card, port) as parameters, the IP address of each port was calculated.

The function reads the values for A, N, S, C, P of each node. As first step in the process, every node calculates (based on identifiers) the IP address of any active port.

The address, after calculation, is represented according to [7] in standard IPv4 format 10.X.Y.Z.

After address calculation the exchange process begins and as a result of its completion, the new IP addresses of each port of every network nodes are shown below in Table 2.

*Table 2*

**Exchanged IP addresses**

| exchanged IP address | A | N | S | C | P |
|---|---|---|---|---|---|
| 10.17.8.78 | 1 | 1 | 2 | 1 | 3 |
| **10.18.4.201** | 1 | 1 | 2 | 1 | 2 |
| **10.20.18.137** | 1 | 1 | 2 | 1 | 1 |
| 10.17.8.66 | 1 | 1 | 2 | 1 | 0 |
| 10.18.4.206 | 1 | 2 | 1 | 3 | 3 |
| 10.18.4.202 | 1 | 2 | 1 | 3 | 2 |
| **10.19.8.75** | 1 | 2 | 1 | 3 | 1 |
| 10.18.4.194 | 1 | 2 | 1 | 3 | 0 |
| 10.19.8.78 | 1 | 3 | 2 | 1 | 3 |
| 10.19.8.76 | 1 | 3 | 2 | 1 | 2 |
| **10.20.12.133** | 1 | 3 | 2 | 1 | 1 |
| 10.19.8.66 | 1 | 3 | 2 | 1 | 0 |
| 10.20.12.142 | 1 | 4 | 3 | 2 | 3 |
| 10.20.12.138 | 1 | 4 | 3 | 2 | 2 |
| 10.20.12.134 | 1 | 4 | 3 | 2 | 1 |
| 10.20.12.130 | 1 | 4 | 3 | 2 | 0 |

A benefit of this approach is that the two addresses of an end-to-end link are always in sequence.

## 6. Conclusions

As a conclusion, the automatic method to generate network addresses based on unique hardware, has the advantages:

- the address calculation and allocation process is automatic;
- the process doesn't require a human operator;
- the process doesn't need any external hardware and software;
- the generated addresses have not random values but arranged according to the algorithm in a predetermined order;

Running the protocol after address allocation process brings advantages:

- the protocol is independent of any upper layer protocol, it runs at OSI layer 2 and doesn't require an upper layer protocols to be established;
- the protocol could run on different physical transport media such as: Ethernet, LAPD, ATM, Frame Relay, etc.;
- the addresses on a direct link between two ports are in sequence;
- based on the opposite port address, the local address is recalculated.

## R E F E R E N C E S

[1] *C.V. Marian, D. Oprea*, Method and system for automatic address allocation in a network and network protocol therefor, United States Patent No. US 7000029, February 2006

[2] *C.V. Marian and D. Oprea*, Method and system for automatic address allocation in a network and network protocol therefor, Canadian Intellectual Property Office Patent: CA 2390331, February 2006

[3] *R. Droms*, Dynamic Host Configuration Protocol, IETF RFC 2131, March 1997.

[4] *R. Droms, J. Bound, B. Volz*, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), IETF RFC 3315, December 1995

[5] *S. Deering, R. Hinden*, Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 2460, December 1998

[6] *Darpa*, DOD Standard Internet Protocol, IETF RFC 760, January 1980

[7] *Darpa*, Internet Protocol - DARPA Internet Program Protocol Specification, IETF RFC 791, September 1981

[8] *C.V. Marian, V. Croitoru, and D. Oprea*, "Proposed Method of Automatic IP Addresses Calculation and Allocation", ISETC 2012 (IEEE 10th International Symposium on Electronics and Telecommunications) Proceedings, Timisoara, Romania, pp 107-110, IEEE Catalog Number CFP1203L-PRT, ISBN: 978-1-4673-1174-8, November 2012

[9] *A. Retana, R. White, V. Fuller*, Using 31-Bit Prefixes on IPv4 Point-to-Point Links, IETF RFC 3021, December 2000

[10] *C.V. Marian, V. Croitoru, and D. Oprea*, "Proposed Communication Protocol between Network Nodes and Address Exchange", ISSCS 2013 (IEEE 11th International Symposium on Signals, Circuits and Systems) Proceedings, Iasi, Romania, pp 116/1-4, IEEE Catalog Number CFP13816-CDR, ISBN: 978-1-4673-6141-5, July 2013.