...the broadest narrowband money can buy

RADIO DATA NETWORKS

# Ethernet in MORSE

# MORSE Guide 3

version 7.45
9/25/2012

**RACOM s.r.o.** • Mirova 1283 • 592 31 Nove Mesto na Morave • Czech Republic
Tel.: +420 565 659 511 • Fax: +420 565 659 512 • E-mail: racom@racom.eu

## Table of Contents

1. Ethernet in MORSE ..... 5
   1.1. Configuration overview ..... 6
2. M-IP-M mode ..... 10
   2.1. M-IP-M example 1 ..... 10
   2.2. M-IP-M example 2 ..... 13
3. IP-M-IP mode ..... 15
   3.1. IP-M-IP example 1 ..... 15
   3.2. IP-M-IP example 2 ..... 19
   3.3. IP-M-IP multicast ..... 23
4. MAS mode ..... 25
   4.1. MAS example 1 ..... 25
   4.2. MAS example 2 ..... 28
   4.3. MAS example 3 ..... 31
5. Connection via MORSE Application Server ..... 33
6. Format of UDP datagram IPGW for Morse ..... 43

## List of Figures

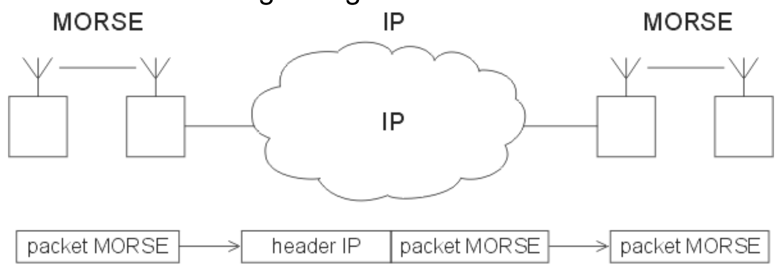
1.1. Ethernet channel modes ..... 8
1.2. Conversion of addresses for UDP ..... 9
2.1. M-IP-M example 1 ..... 10
2.2. M-IP-M example 2 ..... 13
3.1. IP-M-IP example 1 ..... 15
3.2. IP-M-IP example 2 ..... 19
3.3. IP-M-IP example 3 ..... 23
4.1. MAS example 1 ..... 25
4.2. MAS example 2 ..... 28
4.3. MAS example 3 ..... 31

# 1. Ethernet in MORSE

The Ethernet channel in the MORSE CU offers 3 modes of operation, M-IP-M, IP-M-IP, MAS:

### M-IP-M

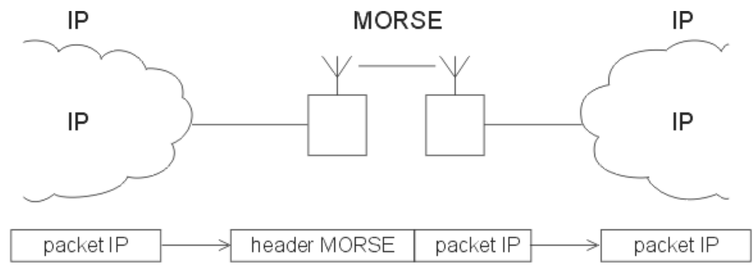
The packet travelling through the MORSE network has a section going along the IP network inserted into its path. The packet has the following configuration:




Looking at the configuration of the MORSE network the IP section appears as one of the routing jumps between two nodes. Routing tables are compiled in the same way as if this section would be one of the radio jumps. The relationship with IP is configured in the Ethernet channel and Art tables in two CUs at the interface with the IP part.

### IP-M-IP

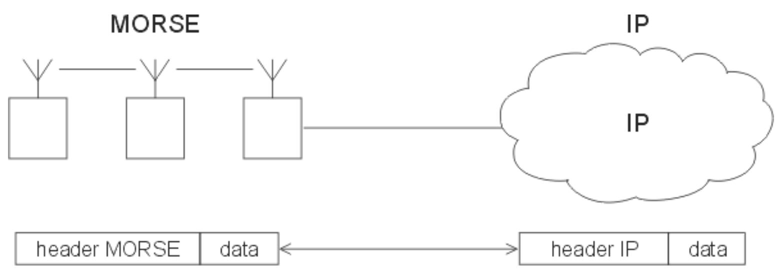
The packet in the IP network has a section going along the MORSE network inserted into its path. Packet configuration:




For the MORSE network the event begins with receipt of the packet from the user output of the Ethernet channel to the node and ends with transfer of the packet through the user output of the last node to the Ethernet channel. The relationships with IP are contained in the Ethernet channel and Art tables in the end CU.

### MAS

The MORSE Application Server is designed for connecting an application running on the LAN network with end points lying in the MORSE network. MAS replaces the IP header with the MORSE header and vice versa:



If the application in the LAN network sends a UDP datagram to any of the terminals in the MORSE network then the MORSE destination address is contained in the data of this datagram and is used by MAS to compile the MORSE header. The "source" MORSE address as the second main parameter of the MORSE header is derived from the senders IP address and port by means of Art. In a similar manner the IP destination address and port number are derived from the MORSE destination address when processing the incoming packet from the MORSE network. The MORSE source address is then inserted into the MORSE pseudoframe in the data part of the UDP datagram.

A description of these modes is given in the Ethernet[1] chapter of MORSE firmware documentation.

## 1.1. Configuration overview

The following diagram *Ethernet channel modes* provides a clear portrayal of the principles of configuration for these three modes. It does not compensate for a detailed description. In order to place all important information into one picture the following symbols and simplifications were used:

- At the top there is the configuration for mode M-IP-M, in the middle for IP-M-IP, and at the bottom for MAS.
- The packet´s path begins with the mode symbol, for example : M-IP-M
- Symbology derived from the Communication unit chapter is used here. The squares in the middle represent nodes in the CU, rectangles are ETH channels in the CU, enlarged so that the respective operations can be marked inside them.
- Each picture illustrates two CUs. For example, in the upper picture the packet arrives from the MORSE network over the RFC channel via two nodes to the ETH channel. It then gets to the second CU through the "IP network", which continues on the left side of the picture, and from here it continues via ETH, nodes, and RFC to the MORSE network. The right ETH contains the operations when passing from the node to LAN, and on the left when passing from LAN to the node.

### Description of the function of individual modes

### M-IP-M

A packet arrives from the MORSE network. It enters the first node and according to routing continues to the second whose network output N is connected to the ETH channel. Routing in the second node determines the address of the following node **to**, with which the packet enters the ETH channel.

Here **IPdst**,which appertains to the MORSE ETH channel on the second side of LAN, is found in the Art table according to the **to** address. A UDP datagram is created which contains the original MORSE packet. If in the range of mask **(n)et mask** the **IPdst** is the same as the actual IP address of the ETH channel **MyIP**, then the UDP datagram is sent to **IPdst**. If it is not the same it is sent to the **(g)ateway** address.

After passing through the IP network the UDP datagram enters the ETH channel with address **IPdst** (left side of the diagram). After removing the IP header the original MORSE routing packet appears which passes through the retranslation output **r** from the ETH channel to the node with address **to**. It continues according to routing to the neighbouring node, through its output N to the RFC channel and on through the MORSE network.

[1] http://www.racom.eu/eng/support/firmware/mr400/eth.html

### IP-M-IP

A datagram arrives from the IP network to the ETH channel. According to the **IPdst** address the MORSE destination address dst is created in one of three ways:
- conversion according to **Art** table EPe0tI
- if IPdst is not found in the table and if IPdst is the same as base EPe0ta in the range of the inverse **mask** EPe0tm, then the MORSE dst is made up from MyMORSE and IPdst according to mask EPe0tm
- if none of these procedures are possible **default gw** from Art table EPe0tI is used

The resulting address **MORSE dst** is used for compiling a user type MORSE packet which is transferred to the node. From there the packet continues through the MORSE network according to routing rules. Once the packet arrives at the node with the MORSE address dst it is transferred from this node through the user output to the ETH channel.

Here the MORSE header is removed leaving the original IP datagram. This is sent to the IP network in the normal manner, i.e. to address **IPdest** or to address EPe0g, i.e. **(g)ateway**

### MAS

A) a UDP frame arrives from the IP network. An item labelled here as **"gw"** is compiled from its parts **port src**, the lower part of address **IP src** and two other bits. The **MORSE src** address which appertains to the IP application which sent the UDP frame is found in the Art table EPe0ts according to this. MAS compiles a MORSE packet supplied with this MORSE src address. The MORSE dst address and transferred data are taken from the incoming UDP frame.

This user type packet is sent to the node and continues from there through the MORSE network to the destination CU.

B) A packet arriving from the MORSE network can have a destination address the same as the MAS MORSE and then it is sent through the user output to the ETH channel. Often **Multiaddressing** is used, whereby the incoming packet, which has a different address to the address of the node, is directed by routing to the link output. From here it is redirected by parameter Ne1MLn to the user output and to ETH.

In the ETH channel the destination address of the MAS MORSE incoming packet is compared with **base** EPe0tB. If it is the same in the range of **mask** EPe0tM the packet is processed in MAS, otherwise it continues in mode IP-M-IP. According to the MAS MORSE address item **"gw"** is found in Art table EPe0ts. A UDP frame is compiled from this item, the actual IP address, and data from the incoming packet, and is sent to the connected LAN.

Conversion of UDP datagram addresses to MORSE and back is clearly illustrated in the diagram *Conversion of addresses for UDP*. A more detailed description of the UDP datagram is contained in article Chapter 6, *Format of UDP datagram IPGW for Morse*.

### Mode selection

Channel ETH can operate in two or three modes at the same time. For activation of a mode it is necessary to insert a non-zero Art table number in the EPe0t menu.

Diagrams are available separately in PDF format: Ethernet channel modes[2], Conversion of addresses for UDP[3].

[2] http://www.racom.eu/images/radost/images/morse/mas-mode.pdf
[3] http://www.racom.eu/images/radost/images/morse/mas-adr.pdf

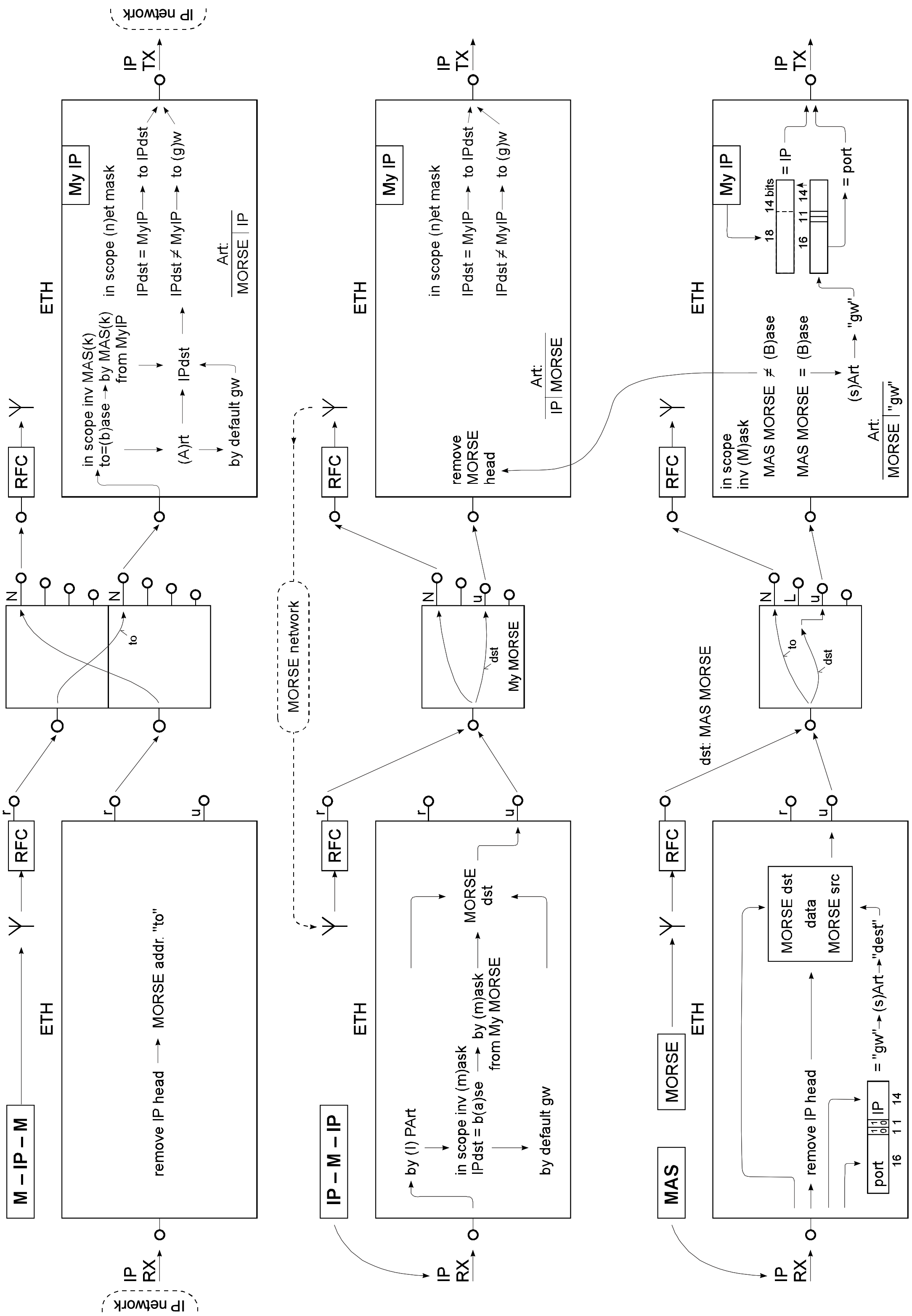



*Fig. 1.1: Ethernet channel modes*

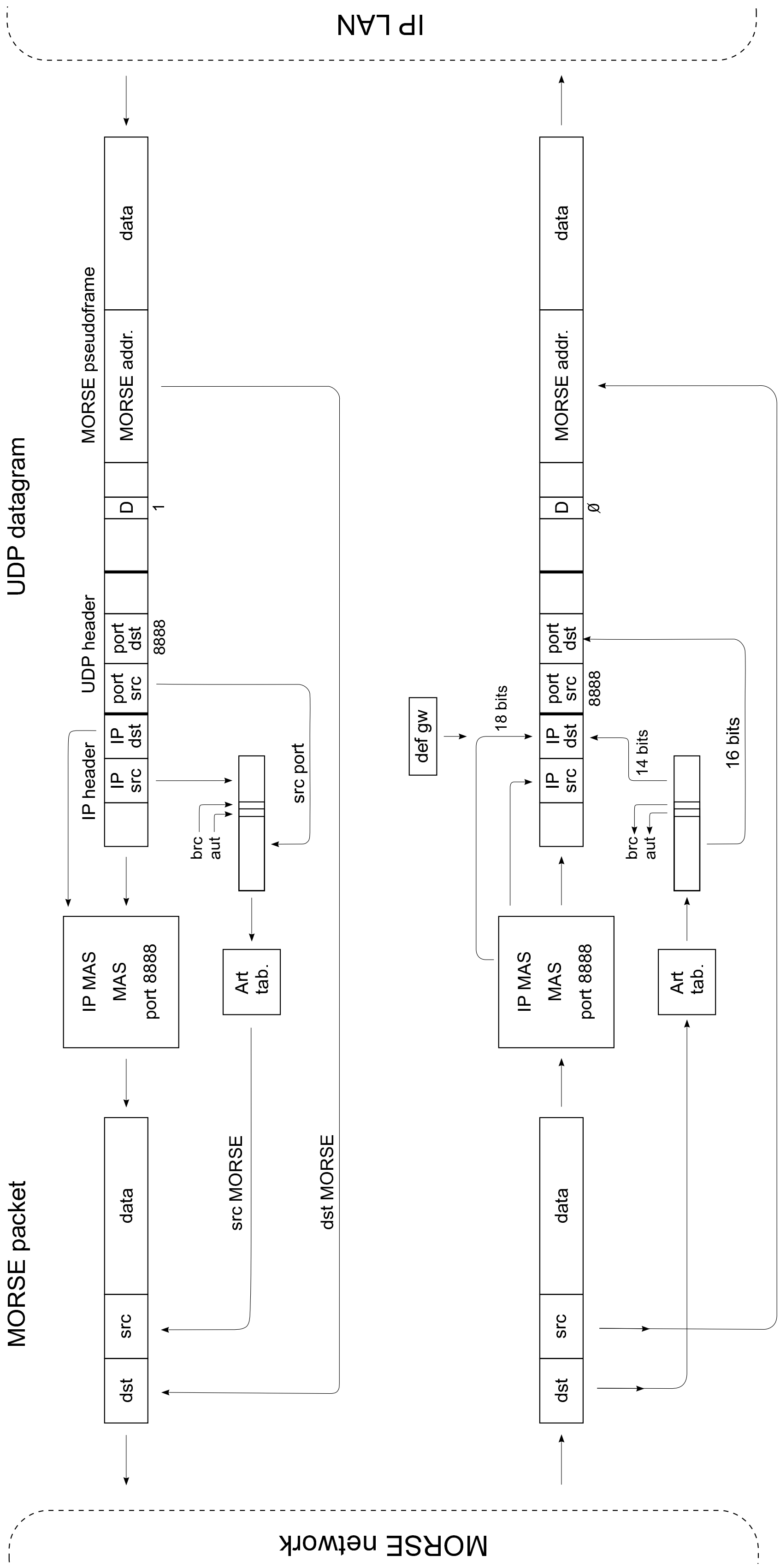



*Fig. 1.2: Conversion of addresses for UDP*

# 2. M-IP-M mode

**Note**

In next diagrams there is used the symbolism derived from chapter Communication Unit. For example the node 690F8101 in next picture is connected in menu Ne with it's net output N to the channel ETH0 and the channel ETH0 is connected in menu Ele with it's retranslation output r to the node 690F8101.

## 2.1. M-IP-M example 1

### Connection of CU via ETH

The connection of two MR400 by Ethernet link is used for the example of the M-IP-M mode.

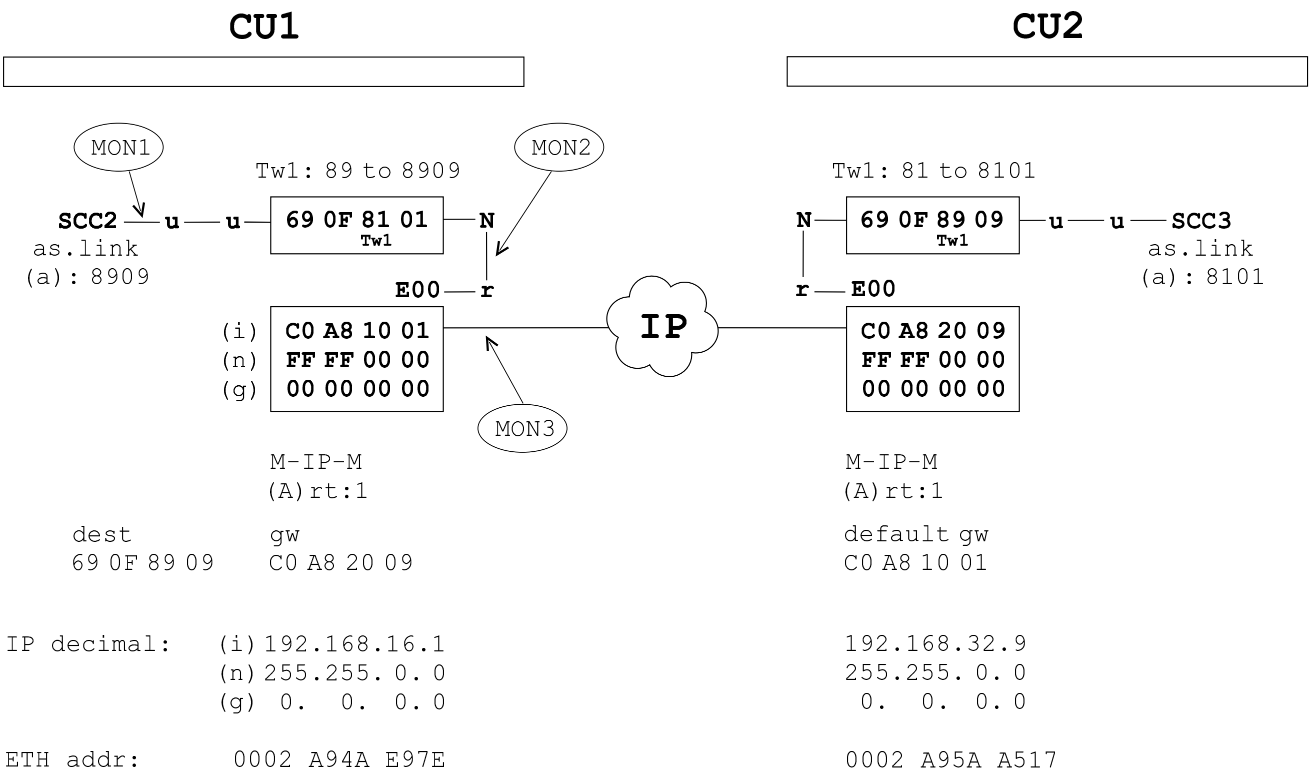



*Fig. 2.1: M-IP-M example 1*

The packet AAAA was received in CU1 by SCC2 port and was sent by Ethernet link to CU2. The monitoring in MON1 to MON3 points:

```
>>
     ...MON1...
CNI mon      |toa       frm      |dst       src      |          size|TT N
13:05:04.588|                    |00008909 00008101|S02I   OUT    2||89 0user
AAAA

     ...MON2...
13:05:04.588|690F8909 690F8101|690F8909 690F8101|E00I    IN    2N89 2dat
```

```
AAAA

     ...MON3...
13:05:04.588 rsi:68 tx|0002A95AA517 |0002A94AE97E | IP/UDP/MOR/RET/DAT
0002 A95A A517 0002 A94A E97E 0800 4500 0036 0072 4000 4011 88EA C0A8 1001
C0A8 2009 22B8 22B8 0022 EF70 D200 1391 690F 8909 690F 8101 0A89 690F 8909
690F 8101 AAAA C654

13:05:04.589 rsi:58 rx|0002A94AE97E |0002A95AA517 | IP/UDP/MOR/RET/CTL/ACK
0002 A94A E97E 0002 A95A A517 0800 4500 002C 000B 4000 4011 895B C0A8 2009
C0A8 1001 22B8 22B8 0018 28D0 C100 1391 D200 690F 8101 690F 8909 5D66
```

**Comment:**

1. The packet AAAA was received by SCC2 port with async.link protocol, was equipped by destination address 00008909 and was sent to the node 690F8101, see MON1.
2. Morse packet with address "to" 690F8909 is sent by "net" output to the channel E00, see MON2.
3. The packet is proceed according to M-IP-M. In the table (A)rt1 which is assigned to IP-M-IP mode there is the address IP destination C0A82009 found according to address "to" 690F8909.
4. IP destination is in range (n)et mask FFFF0000 equal to the own IP address C0A81001, that is why the packet is sent to IP destination. The packet goes from the eth.address 0002A94AE97E to 0002A95AA517, from IP address C0A81001 to C0A82009, see MON3.
5. In CU2 there is generated the asknowledge IP/UDP/MOR/RET/CTL/ACK and is sent back to CU1.
6. The packet receiver E00 in CU2 removes the IP head and sends the packet through retranslation output to the node. In this way is finished the delivery to the MORSE address "to" and the packet can continue according to MORSE routing.

**Notes:**

1. If conformity is not found in point 4) the packet is sent to the IP address given in parameter (g)ateway. For example, after changing (n)et mask in CU1 to value FFFFFF00 it is necessary to amend (g)ateway to C0A82009, in order to achieve a connection. This procedure is used for differentiating between packets directed to the local IP network and those to remote networks.
2. By selecting table Art in menu EPe0t one or more of the selected modes (M-IP-M, IP-M-IP or MAS) are activated. The Art Tables extent is limited. The tables Art1 to Art4 together can contain at the most 252 items. It is recommended use less then a 100 items in one table.
3. It is possible to use `default gw` in the Art table in the normal manner for addresses which are not contained in the table, see CU2.
4. Processing of addresses therefore proceeds as follows :
   - Using MORSE routing the address `to` is found for the packet with the help of rTab
   - Upon entering the ETH channel the destination IP address is found
     - by masking, see M-IP-M example 2

    - or in the Art table
    - or from default gw it Art table
- According to `(n)et mask` the packet is sent to the above-mentioned IP address or to `(g)ateway`
- After reaching the destination IP address the IP header is removed from the packet and the packet is transferred to the node with MORSE address `to` and continues through the MORSE network

5. After an idle period longer than is set by parameter `(A)RP ttl` in menu `EPe0P` an exchange of ARP packets takes place first and monitoring then looks as follows:

```
>>
14:30:20.089|                 |00008909 00008101|S02I   OUT    2||89 0user
AAAA
14:30:20.089|690F8909 690F8101|690F8909 690F8101|E00I    IN    2N89 3dat
AAAA
14:30:20.090 rsi:60 tx|FFFFFFFFFFFF |0002A94AE97E | ARP/REQ
FFFF FFFF FFFF 0002 A94A E97E 0806 0001 0800 0604 0001 0002 A94A E97E
C0A8 1001 0000 0000 0000 C0A8 2009 D3DC E2DB DD64 4E75 44AA A704 B917
F327 1537
14:30:20.090 rsi:68 rx|0002A94AE97E |0002A95AA517 | ARP/REP
0002 A94A E97E 0002 A95A A517 0806 0001 0800 0604 0002 0002 A95A A517
C0A8 2009 0002 A94A E97E C0A8 1001 D3DC E2DB DD64 4E75 44AA A704 B917
F327 1537 BDE4 7A14 85D2 5B34
14:30:20.090 rsi:68 tx|0002A95AA517 |0002A94AE97E | IP/UDP/MOR/RET/DAT
0002 A95A A517 0002 A94A E97E 0800 4500 0036 0073 4000 4011 88E9 C0A8 1001
C0A8 2009 22B8 22B8 0022 8D3C D200 1392 690F 8909 690F 8101 0B89 690F 8909
690F 8101 AAAA 2788
14:30:20.090 rsi:58 rx|0002A94AE97E |0002A95AA517 | IP/UDP/MOR/RET/CTL/ACK
0002 A94A E97E 0002 A95A A517 0800 4500 002C 000C 4000 4011 895A C0A8 2009
C0A8 1001 22B8 22B8 0018 979C C100 1392 D200 690F 8101 690F 8909 EE98
```

The more detailed description is in the paragraph Ethernet of document MORSE Firmware.

## 2.2. M-IP-M example 2

### CU connected to ETH and RFC channels

In mode M-IP-M the network output of the node is connected to channel ETH0. For generation of the connecting radio route we use another node whose network output (N) is connected to channel RFC. The route of the packet through both nodes is defined in the routing table. In the following example the route between four CUs is led through channels radio, ethernet and radio:

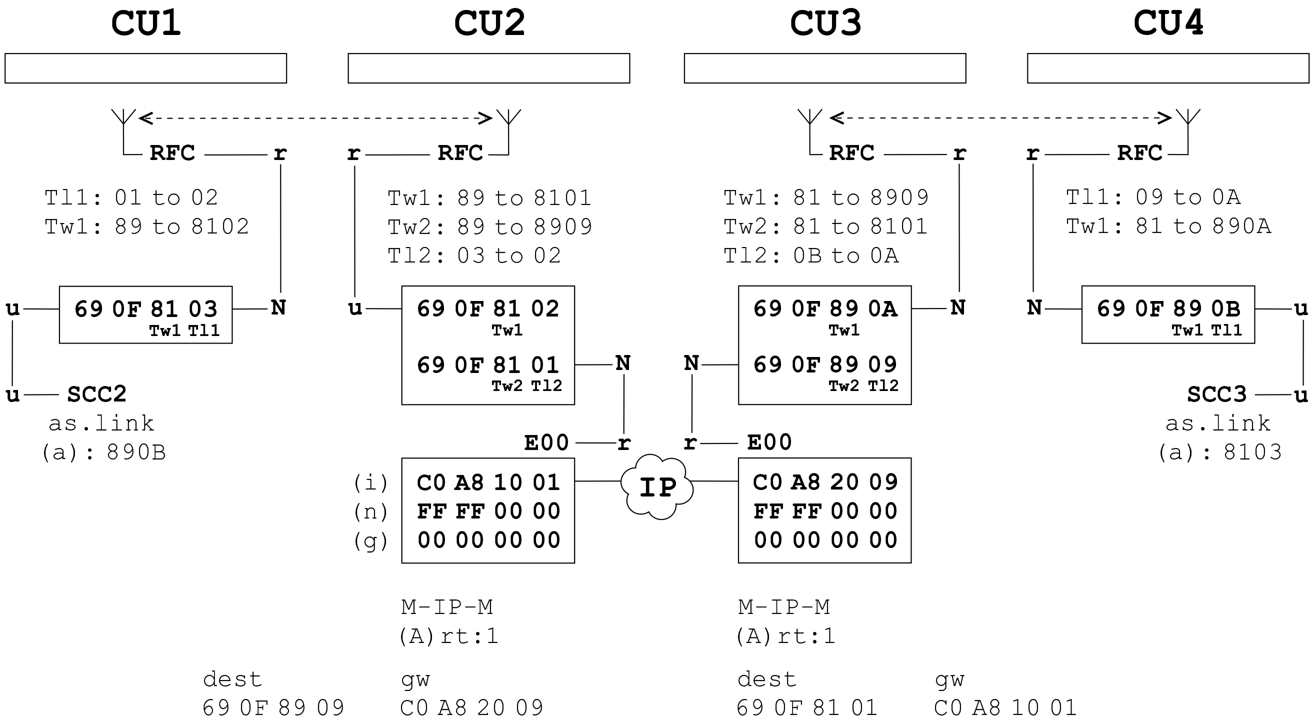



*Fig. 2.2: M-IP-M example 2*

The packet passes from CU1 to CU4 and back as follows:

```
u S00   690F8103    R01
28/ 66  690F8102     -
   -    690F8101    E00
  E00   690F8909     -
   -    690F890A    R01
29/ 67  690F890B   serd

 serd   690F890B    R01
30/ 68  690F890A     -
   -    690F8909    E00
  E00   690F8101     -
   -    690F8102    R01
30/ 69  690F8103  u S00
690F890Bh>
```

An example of monitoring in CU2. The packet is sent from CU1 to CU4 and is monitored on entering CU2 via the radio channel and between node and ETH channel in CU2 :

```
RF mon      |toa      frm      |dst      src      |lNo!DQ!RSS size|TT N
06:47:51.332|690F8102 690F8103|690F890B 690F8103|029*28* 66    2*89 1dat
AAAA
CNI mon     |toa      frm      |dst      src      |          size|TT N
06:47:51.332|690F8909 690F8101|690F890B 690F8103|E00I    IN    2N89 1dat
AAAA
```

### Address translating by mask:

The address of IP destination is sought in table Art according to MORSE address "to". If these addresses are suitably selected it is possible to use mask conversion of addresses instead of using Art. Example 2 is with modified IP addresses:

```
     CU2             CU3
   690F8101        690F8909
        |               |
   C0A80101---IP---C0A80909
   FFFFF000        FFFFF000
```

ETH parameter settings in CU2 and CU3:

```
M-IP-M:
  (A)rt:0; write (E)nable:ON
  (b)ase:690F8000  MAS(K):00000FFF  s(h)ift:0000 ->set Security off!
  (r)epeats:0000  Sec(u)rity:OFF (t)imeout:0  (p)roxy timeout:0s
  (f)rag size:1400bytes  (g)lue (append) up to:0packets
```

When using a mask:

- (b)ase and opposite MORSE address are the same in those parts where the MAS(K) has zeroes
- IP addresses of CU1 and CU2 are the same in those parts where the MAS(K) has zeroes
- MORSE address and IP address of the CU are the same in those parts where the MAS(K) has ones
- Sec(u)rity:OFF - conversion over Ethernet takes place without confirmation, fragmentation OFF, recomended (f)rag size:1400, e.g. the content of routing table does not go through the network at (f)rag size:400
- table Art is not used, if masking is not possible then takes place the Art conversion
- number of addresses unlimited, the IP channel capacity is well exploited

### Characteristics of Art translating:

- Sec(u)rity:OFF - unsecured transfer, fragmentation OFF, (f)rag size:1400 recomended
- Sec(u)rity:ON - secured transfer, fragmentation according (f)rag size:200 to 1400
- possible around 100 addresses at the most
- worse utilization of IP channel

# 3. IP-M-IP mode

## 3.1. IP-M-IP example 1

### Connection of various IP networks

In this example there is used the link from PC1 by ethernet to ETH channel in CU1, then through RF channel to the CU2 and by ethernet to PC2:

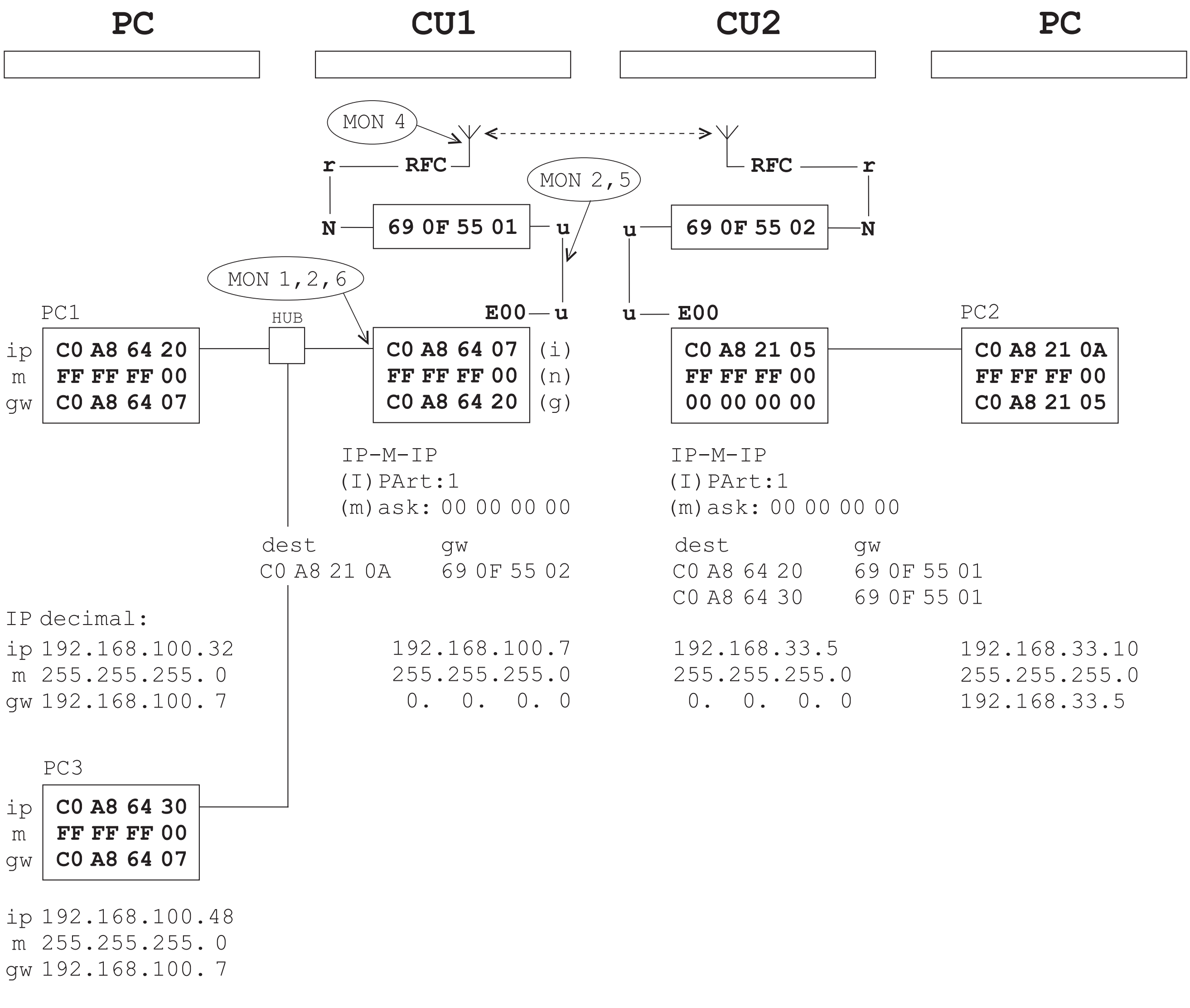



*Fig. 3.1: IP-M-IP example 1*

We can send ping from PC1 to PC2:

```
Start/Run/Open
ping 192.168.33.10 -n 1 -l 2
Enter
```

The way of packet through the channels is showed by monitoring, the labels ...MON... are inserted for better orientation:

```
     ...MON1...
10:39:02.366 rsi:64 rx|FFFFFFFFFFFF |00C09F63CFAB | ARP/REQ
FFFF FFFF FFFF 00C0 9F63 CFAB 0806 0001 0800 0604 0001 00C0 9F63 CFAB
C0A8 6420 0000 0000 0000 C0A8 6407 0000 0000 0000 0000 0000 0000 0000
0000 0000 9696 DA3C
10:39:02.366 rsi:64 tx|00C09F63CFAB |0002A949B897 | ARP/REP
00C0 9F63 CFAB 0002 A949 B897 0806 0001 0800 0604 0002 0002 A949 B897
C0A8 6407 00C0 9F63 CFAB C0A8 6420 0000 0000 0000 0000 0000 0000 0000
0000 0000 9696 DA3C
     ...MON2...
10:39:02.366 rsi:46 rx|0002A949B897 |00C09F63CFAB | IP/ICMP/E_REQ
0002 A949 B897 00C0 9F63 CFAB 0800 4500 0020 00BA 0000 8001 33A8
C0A8 6420 C0A8 210A 0800 E838 0200 4900 6162 6364
     ...MON3...
10:39:02.366|                 |690F5502 690F5501|E00I  OUT   32||89 0user
4500 0020 00BA 0000 8001 33A8 C0A8 6420 C0A8 210A 0800 E838 0200 4900
6162 6364
     ...MON4...
10:39:02.366|690F5502 690F5501|690F5502 690F5501|02E  RFTX    32 89 0dat
4500 0020 00BA 0000 8001 33A8 C0A8 6420 C0A8 210A 0800 E838 0200 4900
6162 6364
10:39:02.530|690F5501 690F5502|690F5501 690F5502|012*29* 80   32*89 0dat
4500 0020 2800 0000 8001 0C62 C0A8 210A C0A8 6420 0000 F038 0200 4900
6162 6364
     ...MON5...
10:39:02.530|                 |690F5501 690F5502|E00I   IN   32|*89 0user
4500 0020 2800 0000 8001 0C62 C0A8 210A C0A8 6420 0000 F038 0200 4900
6162 6364
     ...MON6...
10:39:02.530 rsi:46 tx|00C09F63CFAB |0002A949B897 | IP/ICMP/E_REP
00C0 9F63 CFAB 0002 A949 B897 0800 4500 0020 2800 0000 8001 0C62
C0A8 210A C0A8 6420 0000 F038 0200 4900 6162 6364
```

**Comments on monitoring:**

1. First there is an exchange of packets ARP/REQ and ARP/REP, during which PC1 finds out which ETH address appertains to IP address C0A86407, see MON1.

2. From PC1 to CU1 packet IP/ICMP/E_REQ is sent from ETH address 00C09F63CFAB to 0002A949B897. This packet has IP src C0A8 6420 and IP dst C0A8 210A. The packet is captured by monitoring at the physical input to the ETH channel, see MON2.

3. The ETH channel in CU1 processes it in mode IP-M-IP. In table Art1 address dest C0A8210A is searched for and its respective MORSE address 690F5502 is assigned to the packet, see MON3.

4. This is followed by passage through the MORSE network to CU2 and PC2, where a IP/ICMP/E_REP response, containing the same data, is generated. For passage through the radio channel CU1 see MON4.

5. After passage through the MORSE network the packet is handed over from the node user output 690F5501 to the ETH channel input. Here the packet is processed according to IP-M-IP. The MORSE header is removed from the incoming packet. Furthermore within the scope of mask EPe0

(n)et mask FFFFFF00 compares address IPdest C0A86420 with the actual IP address of the eth. channel C0A86407. Addresses agree and thus the packet is sent to address IPdest; otherwise address EPe0 (g)ateway is used, see MON5.

6. The packet goes from ETH address 0002A949B897 to 00C09F63CFAB, from IP address C0A86407 to C0A86420, see MON6.

### Notes:

1. ETH channel CU1 replies ARP/REP only when the destination address of the ping is not the same as the IP address CU1 in the scope of the mask (in the CU it is EPe 0n). That is why C0A864xx is used in the example on the left side of the IP address and C0A821xx on the right. If another subscriber is present in one of the LAN networks, e.g. PC3 in the example, then its IP address must be the same, in the scope of the mask, as the others, i.e. the same as PC1 and CU1. Then the ETH channel of CU1 does not react to communication between PC1 and PC2.

   This condition complicates the selection of IP addresses. A solution to this problem is given in IP-M-IP example 2.

2. On searching MORSE address destination for the header of the packet in point 3 tools are used in this order:

   ```
   1. searching in (I)PArt table
   2. (m)ask
   3. default gw v (I)PArt
   ```

   1. The destination IP address is searched for in the left column of table Art. If it is found then item `gw` appertains to this item `dest` and is used as the MORSE destination address. The Art Tables extent is limited. The tables Art1 to Art4 together can contain at the most 252 items. It is recommended use less then a 100 items in one table.

   2. If the IP address is not found in the table, then the destination IP address is compared with item EPe0t b(a)se. If they are the same in the part where there are zeros in mask EPe0t (m)ask (in bits), then the MORSE destination is derived from the actual MORSE address. The part of the IP address where the (m)ask has ones is taken and is used to replace the respective part of the MORSE address.

   3. If the IP destination address does not conform to the condition of b(a)se and (m)ask, the MORSE address given in (I)PArt in item `default gw` is used.

   For an illustration of these options `EPe0t` parameters `b(a)se`, `(m)ask` are inserted into the CU1:

```
  INTERNET PROTOCOL GATEWAY:
M-IP-M:
  (A)rt:0; write (E)nable:ON
  (b)ase:00000000  MAS(K):00000000  s(h)ift:0000 ->set Security off!
  (r)epeats:0000  Sec(u)rity:ON  (t)imeout:0  (p)roxy timeout:0s
  (f)rag size:0bytes  (g)lue (append) up to:0packets
IP-M-IP:
  (I)PArt:1
  b(a)se:C0A80400  (m)ask:000000FF
  IP(F)rag. size:552
```

```
MAS:
  (s)Art:0; write (e)nable:ON
  (B)ase:00000000  (M)ask:00000000
```

We insert `default gw to Art`:

```
ART No 1:
items: 1
default gw: 690F5503 (105.15.85.3    )
dest:    gw:
C0A8210A 690F5502 (192.168.33.10   105.15.85.2    )
```

### Address according to (I)PArt

We send a ping to address `C0A8210A` and observe monitoring MON3:

```
ping 192.168.33.10 -n 1 -l 2

12:56:54.985|                 |690F5502 690F5501|E00I   OUT   30||89 0user
                                 ========
4500 001E 0128 0000 8001 333C C0A8 6420 C0A8 210A 0800 349D 0200 6000 6162
                                        ---- ----
```

Address `C0A8210A` is found in table Art and from here address MORSE dest `690F5502` is taken.

### Address according to b(a)se and (m)ask

We send a ping to address `C0A80408`:

```
ping 192.168.4.8 -n 1 -l 2

13:00:09.594|                 |690F5508 690F5501|E00I   OUT   30||89 0user
4500 001E 012B 0000 8001 503B C0A8 6420 C0A8 0408 0800 319D 0200 6300 6162
```

Address `C0A80400` is not found in Art. Its part `C0A804xx` agrees with item `b(a)se C0A804xx` in the scope of the inverse mask `000000FF`. Address MORSE dest is thus made up of part of the actual MORSE address `690F55xx` and part of the IP address `xxxxxx08`, i.e. the result is `690F5508`.

### Address according to default gw

We send a ping to address `C0A80908`:

```
ping 192.168.9.4 -n 1 -l 2

13:05:13.752|                 |690F5503 690F5501|E00I   OUT   30||89 0user
4500 001E 012C 0000 8001 4B3E C0A8 6420 C0A8 0904 0800 309D 0200 6400 6162
```

Address `C0A80400` is not found in Art and does not match the mask. Address MORSE dest is thus taken from item default gw in table Art.

## 3.2. IP-M-IP example 2

### Connection inside one IP network

In menu EPe there is a group of AR(P) parameters, which simplify the design of the IP-M-IP network. Their use brings about a certain risk and this is why it is necessary to proceed very carefully, i.e. see the note at the end of this article. It is possible here to configure a set of IP addresses to which an arp proxy is not sent. The example illustrates the connection of more IP subscribers in the same network with mask 24.

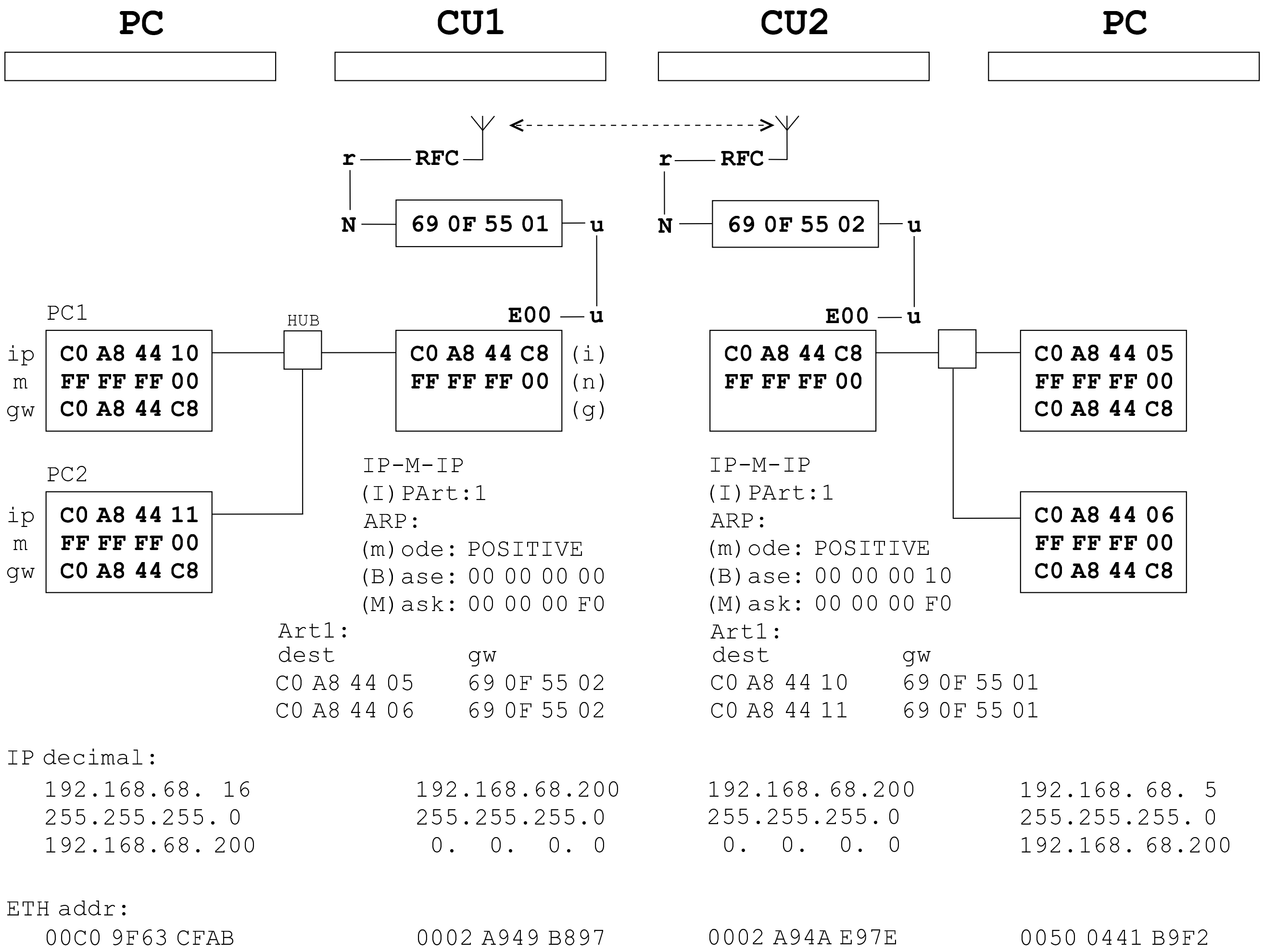



*Fig. 3.2: IP-M-IP example 2*

Members of the IP network PC1, PC2 and others in the left part of the screen can communicate together, if they use addresses in the range 0xC0A84410 to 0xC0A844FE. If any of them calls an address from the range 0xC0A84401 to 0xC0A8440F then CU1 responds ARP/REP and hands over the message to the MORSE network.

If the following ping is sent:

```
ping 192.168.68.17 -n 1 -l 2
```

from PC1 then CU1 does not respond and PC1 and PC2 can communicate without interruption because the destination IP address 192.168.68.17 = 0xC0A84411 is outside the scope defined in menu `EPe0P` parameters:

```
proxy arp (m)ode:POSITIVE
Proxy Arp (B)ase:00000000
Proxy Arp (M)ask:000000F0
```

If the following ping is sent:

```
ping 192.168.68.5 -n 1 -l 2
```

from PC1 then CU1 responds because the destination IP address 192.168.68.5 = 0xC0A84405 is in the range determined by Proxy Arp parameters (and simultaneously IP addresses are the same in the scope of the IP mask). Monitoring of ETH and RFC channels of CU1 then looks as follows:

```
08:14:26.669 rsi:64 rx|FFFFFFFFFFFF |00C09F63CFAB | ARP/REQ
FFFF FFFF FFFF 00C0 9F63 CFAB 0806 0001 0800 0604 0001 00C0 9F63 CFAB
C0A8 4410 0000 0000 0000 C0A8 4405 0000 0000 0000 0000 0000 0000 0000
0000 0000 2629 62E3
08:14:26.669 rsi:64 tx|00C09F63CFAB |0002A949B897 | ARP/REP
00C0 9F63 CFAB 0002 A949 B897 0806 0001 0800 0604 0002 0002 A949 B897
C0A8 4405 00C0 9F63 CFAB C0A8 4410 0000 0000 0000 0000 0000 0000 0000
0000 0000 2629 62E3

08:14:26.670 rsi:44 rx|0002A949B897 |00C09F63CFAB | IP/ICMP/E_REQ
0002 A949 B897 00C0 9F63 CFAB 0800 4500 001E 11EC 0000 8001 1F8D
C0A8 4410 C0A8 4405 0800 849D 0200 1000 6162

08:14:26.670|                 |690F5502 690F5501|E00I   OUT   30||89 0user
4500 001E 11EC 0000 8001 1F8D C0A8 4410 C0A8 4405 0800 849D 0200 1000 6162

08:14:26.670|690F5502 690F5501|690F5502 690F5501|076  RFTX    30 89 2dat
4500 001E 11EC 0000 8001 1F8D C0A8 4410 C0A8 4405 0800 849D 0200 1000 6162

08:14:26.805|690F5501 690F5502|690F5501 690F5502|05C*29* 77   30*89 5dat
4500 001E 2500 0000 8001 0C79 C0A8 4405 C0A8 4410 0000 8C9D 0200 1000 6162

08:14:26.805|                 |690F5501 690F5502|E00I    IN   30|*89 5user
4500 001E 2500 0000 8001 0C79 C0A8 4405 C0A8 4410 0000 8C9D 0200 1000 6162

08:14:26.806 rsi:44 tx|00C09F63CFAB |0002A949B897 | IP/ICMP/E_REP
00C0 9F63 CFAB 0002 A949 B897 0800 4500 001E 2500 0000 8001 0C79
C0A8 4405 C0A8 4410 0000 8C9D 0200 1000 6162
```

### Setting parameters in the example:

Parameters (m)ode, (B)ase and (M)ask in menu EPe0P determine the group of destination IP addresses to whose ARP/REQ CU1 will respond. Other addresses can be used for communication in the local LAN, in this case PC1 and PC2. Apart from the general condition that IP addresses must be the same within the (n)et mask here are some other possibilities:

```
proxy arp (m)ode:POSITIVE
Proxy Arp (B)ase:00000000
Proxy Arp (M)ask:000000F0
```

The destination IP must be the same as the Proxy Arp (B)ase in the scope of Proxy Arp (M)ask; here with regard to (n)et mask `C0A8440x`. CU1 responds to this group of IP addresses with an ARP/REP message and generates a packet for the MORSE network. The second possibility for settings in CU1 is:

```
proxy arp (m)ode:NEGATIVE
Proxy Arp (B)ase:00000010
Proxy Arp (M)ask:000000F0
```

Parameter NEGATIVE determines that ARP/REP is not returned to ARP/REQ with addresses selected in that way and thus may be used for communication in the local LAN.

Option:

```
proxy arp (m)ode:NORMAL
```

disables this function. Then the ETH channel behaves as in the previous example, i.e. responds only to ARP/REQ with IP addresses outside of (n)et mask.

### Poznámky:

1. An exchange of packets ARP/REQ and ARP/REP occurred between ETH addresses PC1 and CU1. However, these packets contain destination address PC2, i.e. C0A8 4405, because this lies inside the network mask PC1.

2. Note that the IP address of the modem is not important here. It doesn´t appear in monitoring and can be the same for all CUs in the network. Default gateway can be the address for operation outside the scope of the IP mask.

3. Upon careful selection of MORSE addresses it is possible to use a mask to generate MORSE addresses. Then the Art table is empty, address CU2 is 690F5505 according to IP address PC2. Routing tables need to be filled and in menu Ne use Multiaddressing.

4. Configuration of CU1 is given in the following overview:

```
Menu Ne:
Nid|address |M | u   s | L   N |l w n g H|sTO Err  Cent vTO hTO
(0) 0049B897     -  S00| -  R00|0 0 0 0 -| 15 SERV  OFF 304  30
(1) 690F5501    E00 S00| -  R01|0 0 0 0 -| 15 SERV  OFF 304  30
(2) 00000000    S01 S00| -  R02|0 0 0 0 -| 15 SERV  OFF 304  30
(3) 00000000    S02 S00| -  R03|0 0 0 0 -| 15 SERV  OFF 304  30
(4) 00000000    S03 S00| -  R04|0 0 0 0 -| 15 SERV  OFF 304  30

 Channel to Node Interface:
    retranslation     |   user+service             lim
 id N  A t          m | N  A t Base     m  sec brc S   e
(0) 0   NO AR         | 1   NO AR          usr OFF NONE

  Internet Protocol:
Eid| ip address            |    net mask         |       gw      |
```

```
(0) C0A844C8 192.168.68.200 FFFFFF00 255.255.255.0 00000000 0.0.0.0

  INTERNET PROTOCOL GATEWAY:
M-IP-M:
  (A)rt:0; write (E)nable:ON
  (b)ase:00000000  MAS(K):00000000  s(h)ift:0000 ->set Security off!
  (r)epeats:0000  Sec(u)rity:ON  (t)imeout:0  (p)roxy timeout:0s
  (f)rag size:0bytes  (g)lue (append) up to:0packets
IP-M-IP:
  (I)PArt:1
  b(a)se:C0A84400  (m)ask:00000000
  IP(F)rag. size:552
MAS:
  (s)Art:0; write (e)nable:ON
  (B)ase:00000000  (M)ask:00000000

ARP:
(A)RP ttl:30s
A(R)P timeout:50ms
proxy arp (m)ode:POSITIVE
Proxy Arp (B)ase:00000000
Proxy Arp (M)ask:000000F0

ART No 1:
items: 1
default gw: 00000000 (0.0.0.0        )
dest:    gw:
C0A84405 690F5502 (192.168.68.5    105.15.85.2    )
C0A84406 690F5502 (192.168.68.6    105.15.85.2    )
```

5. If proxy-arp settings are not configured and the CU is connected to the LAN, then in menu Epe0P it is necessary to set parameter

   ```
   proxy arp (m)ode:NORMAL
   ```

   or both parameters:

   ```
   Proxy Arp (B)ase:00000000
   Proxy Arp (M)ask:00000000
   ```

   to zero. In another case (random content of (B) and (M)) spurious packets may appear in LAN.

## 3.3. IP-M-IP multicast

### Processing of IP frame type multicast

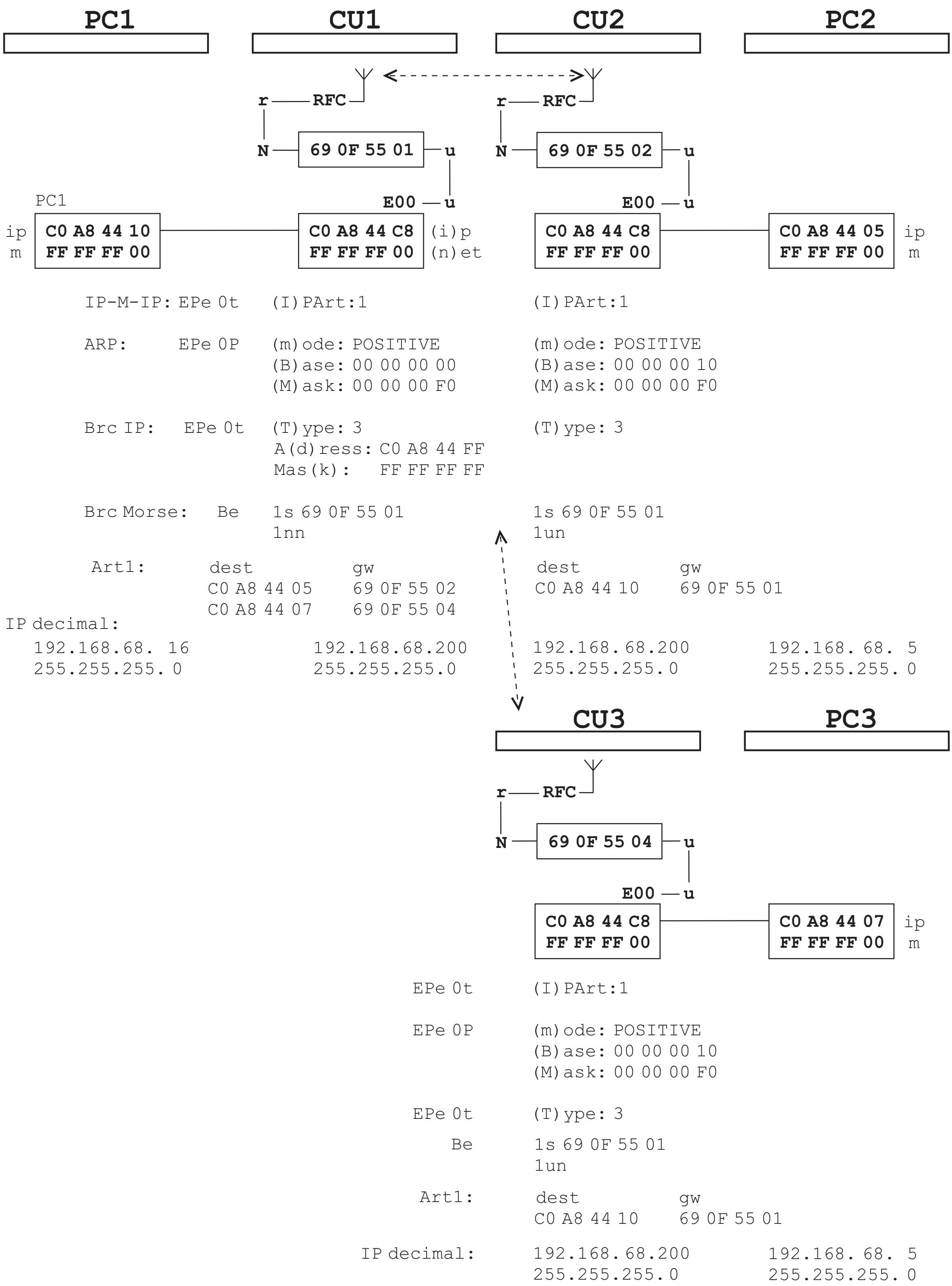



*Fig. 3.3: IP-M-IP example 3*

The example relates to the Example 2 completed by the processing of IP multicast frame sent from the PC1.

The IP multicast is similar to the MORSE broadcast. It can occur in this case, when the PLC master sends the IP frames to all PLCs in the network. The entering MORSE CU then spreads it as the broadcast packet. The outgoing CU generates again the multicast IP frame.

The auxiliary MORSE CU with `EPe 0tP` menu or standard PC serves as the source of multicast IP frame. It sends ping:

```
ping 192.168.68.255 -n 1 -l 2
```

The address of multicast IP frame contains binary one where the netmask contains binary zero.

The parameters in menu `EPe 0t` must be set for right processing of the multicast frame. Set in all CUs `Type: 3`. In the entering CU set more `A(d)dress` a `Mas(k)`, which define the IP address of multicast frame. The outgoing CUs does not use `A(d)dress` a `Mas(k)`. In this example are:

```
A(d)dress:C0A844FF
Mas(k):FFFFFFFF
```

or for example:

```
A(d)dress:000000FF
Mas(k):000000FF
```

All CUs need to have properly set the broadcast menu Broadcast `Be`.

The example of receiving the multicast frame in CU1 and sending the brc MORSE packet in RF channel:

```
13:43:37.979 rsi:50 rx|FFFFFFFFFFFF |0002A95EADEA | IP/ICMP/E_REQ
FFFF FFFF FFFF 0002 A95E ADEA 0800 4500 0024 050D 4000 4001 2B6C C0A8 4410 C0A8 44FF 0800
B3D6 0000 0000 0000 6444 010A DEDA
13:43:37.979|690F5501 690F5501|0D2679F4 690F5501|R01I    IN   36N09 0dat 0brc
4500 0024 050D 4000 4001 2B6C C0A8 4410 C0A8 44FF 0800 B3D6 0000 0000 0000 6444 010A DEDA
```

The broadcast packet goes through MORSE network to CU2 and then is send as multicast frame to the ethernet. Analogical monitoring can be observed also in CU3.

```
13:43:38.024|690F5501 690F5501|000079F4 690F5501|R01I   OUT   36n09 0dat 0brc
4500 0024 050D 4000 4001 2B6C C0A8 4410 C0A8 44FF 0800 B3D6 0000 0000 0000 6444 010A DEDA
13:43:38.024|                 |000079F4 690F5501|E00I    IN   36|*09 0usr brc 0
4500 0024 050D 4000 4001 2B6C C0A8 4410 C0A8 44FF 0800 B3D6 0000 0000 0000 6444 010A DEDA
13:43:38.024 rsi:50 tx|FFFFFFFFFFFF |0002A95B93F9 | IP/ICMP/E_REQ
FFFF FFFF FFFF 0002 A95B 93F9 0800 4500 0024 050D 4000 4001 2B6C C0A8 4410 C0A8 44FF 0800
B3D6 0000 0000 0000 6444 010A DEDA
```

The standard no-multicast frame can be sent also as broadcast when it's dst IP address is written in the `A(d)dress` parameter with mask `FFFFFFFF`. The other combination of address and mask can be used also, e.g. `C0A84400`, `FFFFFF00` with meeting the condition, that the address contains zeros in all bits where the zero is in the mask. The packet of broadcast type outputs the MORSE network as multicast frame.

# 4. MAS mode

## 4.1. MAS example 1

### Connection of a sole application over MAS

Communication unit CU1 with Morse Application Server is connected via this Ethernet channel to IP LAN, in which the PC operates with application Setr. This application communicates over MAS with CU2, or with other CU in the MORSE network.

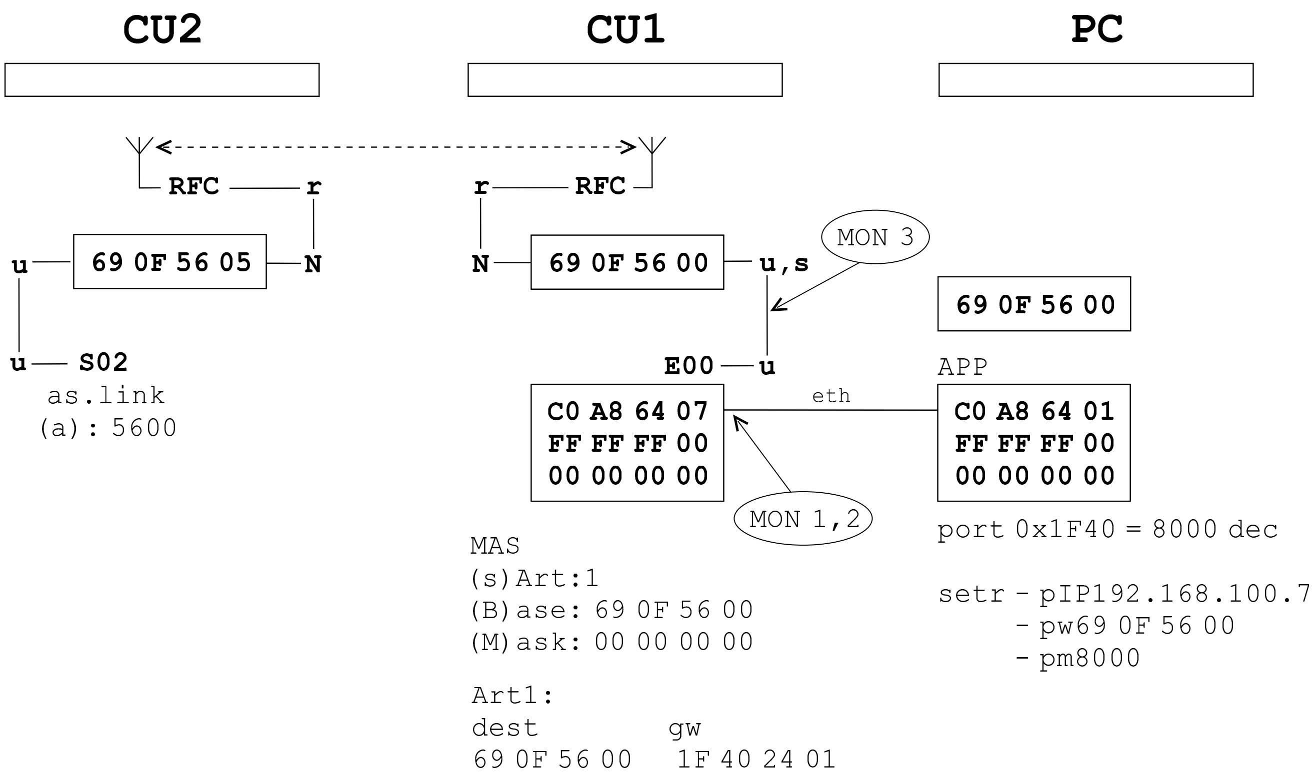



*Fig. 4.1: MAS example 1*

Program Setr here replaces a general IP application. It can be started using command :

```
setr -pIP192.168.100.7 -pw690f5600 -pm8000
```

- `-pIP192.168.100.7` = IP address of MAS
- `-pw690f5600` = MORSE address with which the IP application will appear in the MORSE network
- `-pm8000` = port number of IP application

On starting Setr table Art related to MAS is automatically filled:

```
ART No 1:
items: 1
default gw: 00000000 (0.0.0.0        )
```

```
dest:    gw:
690F5600 1F402401 (105.15.86.0     31.64.36.1     )
```

- 690F5600 = MORSE address related to the IP application
- 1F40 = port number of IP application
- 2401 = lower 14 bits of IP address of application, upper 18 bits are added from the IP address of MAS

We send a test packet from IP application of Setr to CU2 690F5605. Firstly we prepare a destination address:

```
!h690F5605 Enter
```

Then using command `mtU` we send data 0xAAAA and observe in monitoring the transformation from IP format to MORSE format:

```
     ...MON1...
10:19:11.121 rsi:60 tx|FFFFFFFFFFFF |0002A949B897 | ARP/REQ
FFFF FFFF FFFF 0002 A949 B897 0806 0001 0800 0604 0001 0002 A949 B897
C0A8 6407 0000 0000 0000 C0A8 6401 0000 8F01 690F 5600 C07E 800E 43D0
AB0C 1A49
10:19:11.121 rsi:64 rx|0002A949B897 |00C09F63CFAB | ARP/REP
0002 A949 B897 00C0 9F63 CFAB 0806 0001 0800 0604 0002 00C0 9F63 CFAB
C0A8 6401 0002 A949 B897 C0A8 6407 0000 0000 0000 0000 0000 0000 0000
0000 0000 2592 FBA6
     ...MON2...
10:19:11.404 rsi:52 rx|0002A949B897 |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A949 B897 00C0 9F63 CFAB 0800 4500 0026 1295 0000 8011 DED8
C0A8 6401 C0A8 6407 1F40 22B8 0012 8137 0000 8981 690F 5605 AAAA
     ...MON3...
10:19:11.404|                 |690F5605 690F5600|E00I   OUT    2||89 1user
AAAA
```

Source and destination IP addresses C0A86401 and C0A86407, source and destination IP ports 1F40 and 22B8, address MORSE destination 690F5605 and data AAAA are contained in monitoring MON2.

In monitoring MON3 the source MORSE address is assigned to the IP application 690F5600, destination MORSE address 690F5605 and data AAAA.

A test of transmission from MORSE to IP is carried out by sending data 0xBBBB from CU2 690F5605 to address 690F5600, where the IP application is located:

```
08:35:53.893|                 |690F5600 690F5605|E00I    IN    2|*89 5user
BBBB
08:35:53.893 rsi:52 tx|00C09F63CFAB |0002A949B897 | IP/UDP/MOR/USR/DATA
00C0 9F63 CFAB 0002 A949 B897 0800 4500 0026 0161 4000 4011 F00C
C0A8 6407 C0A8 6401 22B8 1F40 0012 70A2 0000 8905 690F 5605 BBBB
```

### Notes on configuration:

1. In menu `EPe0t` it is defined by means of parameters

   ```
   MAS:
     (s)Art:1; write (e)nable:ON
     (B)ase:690F5600  (M)ask:00000000
   ```

   that MAS works only with application 690F5600.

2. Table Art contains the conversion between the MORSE address of the application in column dest and the pair IP port/IP address of the application. From the IP address only the 14 lower bits are contained here, the others are taken from the actual IP address of MAS.

3. Art table can be filled automatically as in the given example or manually.

4. The Art Tables extent is limited. The tables Art1 to Art4 together can contain at the most 252 items. It is recommended use less then a 100 items in one table.

## 4.2. MAS example 2

### Connection of more applications over MAS

By using parameters (B)ase, (M)ask and the Multiaddressing function MAS can work with more IP applications which communicate with various CUs in the MORSE network.

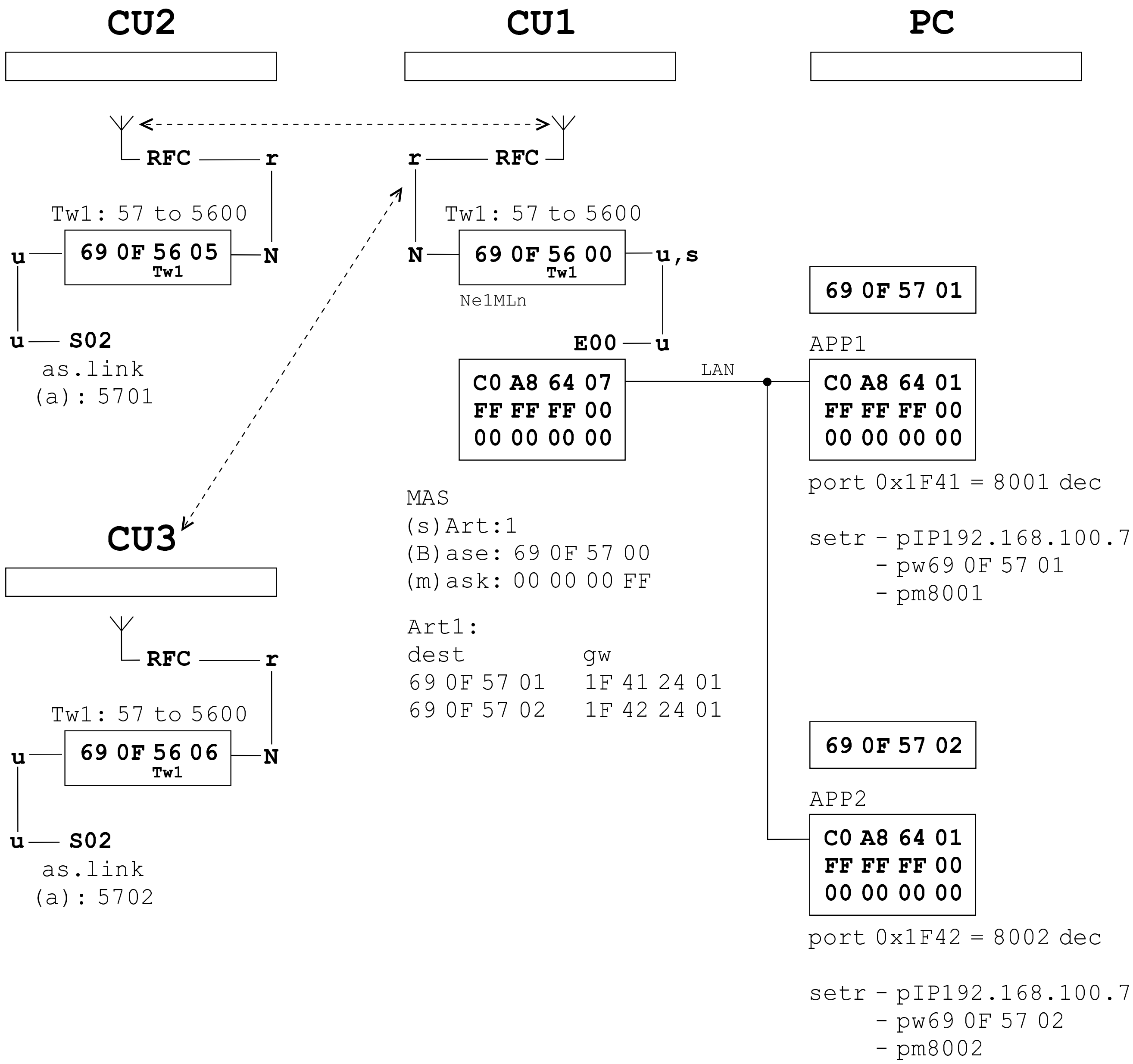



*Fig. 4.2: MAS example 2*

Two applications in the example are located in a common PC. They differ by the number of the IP port and the MORSE address:

```
setr -pIP192.168.100.7 -pw690f5701 -pm8001

setr -pIP192.168.100.7 -pw690f5702 -pm8002
```

Art table contains a translation for each application:

```
ART No 1:
items: 2
default gw: 00000000 (0.0.0.0        )
dest:    gw:
690F5702 1F422401 (105.15.87.2     31.66.36.1     )
690F5701 1F412401 (105.15.87.1     31.65.36.1     )
```

A test packet with data 0xAAAA is sent from the application 690F5701 to CU2 690F5605 using command `mtU`:

```
12:57:15.026 rsi:52 rx|0002A949B897 |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A949 B897 00C0 9F63 CFAB 0800 4500 0026 0130 0000 8011 F03D
C0A8 6401 C0A8 6407 1F41 22B8 0012 8134 0000 8983 690F 5605 AAAA
12:57:15.026|                 |690F5605 690F5701|E00I   OUT    2||89 3user
AAAA
```

A packet with data ASCII aaaa, i.e. 0x61616161 is sent from application 690F5701 to CU3 690F5606 using command `mtu`:

```
13:06:57.921 rsi:54 rx|0002A949B897 |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A949 B897 00C0 9F63 CFAB 0800 4500 0028 0134 0000 8011 F037
C0A8 6401 C0A8 6407 1F41 22B8 0014 6916 0000 8984 690F 5606 6161 6161
13:06:57.921|                 |690F5606 690F5701|E00I   OUT    4||89 4user
6161 6161
```

Data 0xBBBB is sent from CU2 690F5605 to address 690F5701:

```
13:02:09.439|                 |690F5701 690F5605|E00I    IN    2|*89 4user
BBBB
13:02:09.439 rsi:52 tx|00C09F63CFAB |0002A949B897 | IP/UDP/MOR/USR/DATA
00C0 9F63 CFAB 0002 A949 B897 0800 4500 0026 0022 4000 4011 F14B
C0A8 6407 C0A8 6401 22B8 1F41 0012 70A2 0000 8904 690F 5605 BBBB
```

Data 0xCCCC is sent from CU3 690F5606 to address 690F5702:

```
13:03:56.078|                 |690F5702 690F5606|E00I    IN    2|*89 5user
CCCC
13:03:56.078 rsi:52 tx|00C09F63CFAB |0002A949B897 | IP/UDP/MOR/USR/DATA
00C0 9F63 CFAB 0002 A949 B897 0800 4500 0026 0024 4000 4011 F149
C0A8 6407 C0A8 6401 22B8 1F42 0012 5F8E 0000 8905 690F 5606 CCCC
```

### Notes on configuration:

1. In menu `EPe0t` by means of parameters

```
MAS:
  (s)Art:1; write (e)nable:ON
  (B)ase:690F5700  (M)ask:000000FF
```

a group of MORSE addresses is defined for which MAS carries out a conversion of addresses between MORSE and IP. These are addresses 690F5700 to 690F57FF.

2. Each of these addresses has a respective item in the conversion table Art.

3. Packets with these destination addresses arrive from the MORSE network via the standard method of routing to CU1 690F5600. Here, via item,

```
Wide retab. No 1
57to:5600
```

they are directed to the Link node output.

4. Command `Ne1MLn` is used for switching on the Multiaddressing function which directs all packets from the `Link` output to the `user` output. In this way packets for all serviced IP applications get to the Ethernet channel where MAS converts them to IP format.

5. The service output from node 690F5600 must also be directed to E00. If it remains in S00 then service responses are not sent to MAS.

```
Nid|address |M | u   s | L   N |l w n g H|sTO Err  Cent vTO hTO
(0) 0049B897     -  S00| -  R00|0 0 0 0 -| 15 SERV  OFF 304  30
(1) 690F5600 L  E00 E00| -  R01|0 1 0 0 -| 15 SERV  OFF 304  30
(2) 00000000    S01 S00| -  R02|0 0 0 0 -| 15 SERV  OFF 304  30
(3) 00000000    S02 S00| -  R03|0 0 0 0 -| 15 SERV  OFF 304  30
(4) 690F0000    S03 S00| -  R04|0 0 0 0 -| 15 SERV  OFF 304  30
```

## 4.3. MAS example 3

### Applications and MAS lie in different IP networks

If the IP address of the application and MAS differ in their upper 18 bits we use default gw in table Art:

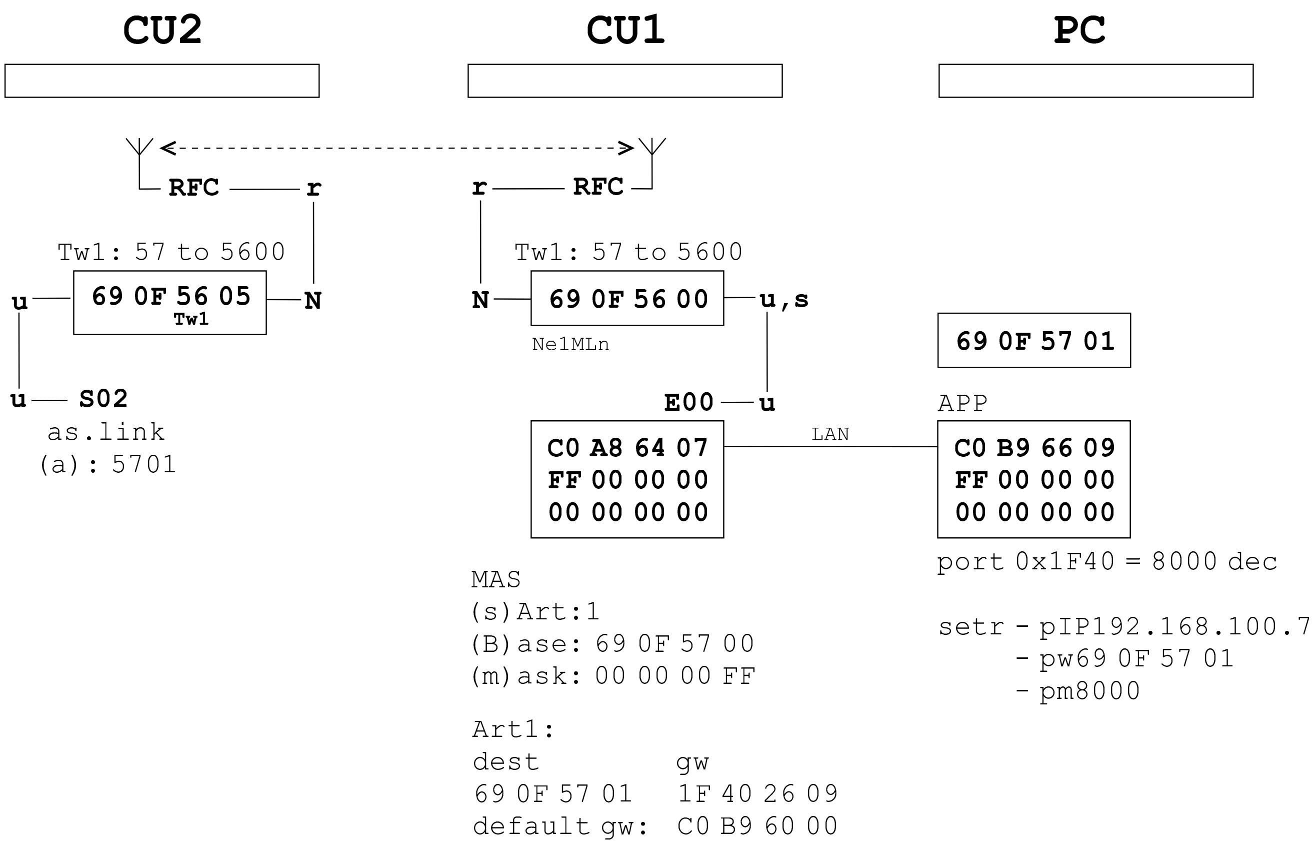



*Fig. 4.3: MAS example 3*

```
setr -pIP192.168.100.7 -pw690f5701 -pm8000
```

```
ART No 1:
items: 1
default gw: C0B96609 (192.185.102.9 )
dest:    gw:
690F5701 1F402609 (105.15.87.1     31.64.38.9    )
```

```
10:54:38.630 rsi:54 rx|0002A949B897 |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A949 B897 00C0 9F63 CFAB 0800 4500 0028 023E 0000 8011 ED14
C0B9 6609 C0A8 6407 1F40 22B8 0014 6702 0000 8981 690F 5605 6161 6161
10:54:38.630|                 |690F5605 690F5701|E00I   OUT    4||89 1user
6161 6161
```

### Notes:

1. IP addresses agree only in the upper 8 bits and for that reason larger masks FF000000 are used.

2. If `default gw` in table Art is not zero then it is used instead of the MAS IP address for generation of the IP address of the application. `Default gw` must agree with the IP address of connected applications in the upper 18 bits. Lower bits are of no significance. It is thus possible to insert the whole IP address of certain application into `default gw`.

### Default address for the direction from IP to MORSE

A packet sent from an application via MAS to the MORSE network doesn´t have a pre-defined port in certain cases. Therefore it is possible to define a default MORSE address for an application, which is searched for in the Art table in three steps:

1. After unsuccessful searching in column "gw" of table Art

2. Address "00000000", which corresponds to item "gw", is searched for in column "dest", e.g. 1F492401, where 1F49 is the selected standard number of the port and 2401 is the lower part of the IP address of the application. This pair

   ```
   00000000 1F492401
   ```

   must be located at the end of Art so that the pair according to point 3. e.g.:

   ```
   690F5709 1F492401
   ```

   lies above it.

3. This item 1F492401is then searched for in column "gw" and its respective item in column "dest" 690F5709 then becomes the MORSE address of the source in the sent packet.

4. Example of a filled in Art table:

   ```
   ART No 1:
   items: 4
   default gw: 00000000 (0.0.0.0         )
   dest:    gw:
   690F5707 1F472401 (105.15.87.7     31.71.36.1     )
   690F5701 1F412401 (105.15.87.1     31.65.36.1     )
   690F5709 1F492401 (105.15.87.9    31.73.36.1     )
   00000000 1F492401 (0.0.0.0         31.73.36.1     )
   ```

5. If the Art table is filled in we can forbid further automatic entries to the table in menu `EPe0t` using parameter `(e)nable:OFF`:

   ```
   MAS:
     (s)Art:1; write (e)nable:OFF
     (B)ase:690F5700  (M)ask:000000FF
   ```

# 5. Connection via MORSE Application Server

Situation - connection of AAA application in Windows through MAS with CU (Communication Unit) MR400 and to the MORSE network. The AAA application is replaced here by Setr.

MC100, MG100, MR900, MR25ET, MCM302ET or sw Walrus and Morce work in the same way as MR400.

Configuration consists of the following steps:

1. Entering IP addresses - into PC and through service cable to CU
2. Ping testing - ping PC -> CU along Ethernet link
3. Starting Setr - CU connected via Ethernet link, ping PC <- CU
4. Setting routing - configuration for a single PC or for more users
5. Setting MAS - select range of PC addresses, filling Art table
6. Test from mtu menu - sending UDP datagram from Setr via MAS to remote CU, sending MORSE packet from CU via MAS to PC
7. Starting application - after this activation simply replace Setr with another AAA application and the connection is ready

The numbered steps are described below in detail, with a brief summary given at the end.

## 1. Local connection via Ethernet

First set a suitable IP address and mask in a PC running Windows98 or WindowsXP, for example:

```
IP adresa - 10.0.0.1
maska     - 255.255.255.0
```

The MORSE CU of the MR400 is in its default state. Connect to the CU using the service cable and set the IP address and mask, e.g.:

```
EPe 0i0A000002    ... to je 10.0.0.2
    0nFFFFFF00    ... to je 255.255.255.0
```

Insert the MORSE address in CU (since version 9.32 this address can stay zero):

```
Ne 1a690F5600     ... MORSE address
```

## 2. Ping

Now we can remove the service cable and connect the PC and CU using the crossed Ethernet cable. Test the connection by sending a ping from the command line in the PC to the MR400:

```
ping 10.0.0.2
```

## 3. Setr

If the ping works we can start Setr in the PC running Windows:

```
setr -pIP10.0.0.2
```

We can test this PC - CU connection in local mode by calling certain services, e.g.:

```
sts    ... which replies with the number of seconds
           since starting the CU, e.g.:
>>312
```

Now it is possible to test a ping from the CU to the PC:

```
EPe 0tP t0A000001   ... IP address of PC 10.0.0.1
        s           ... (s)tart
        e           ... r(e)port gives the result
        p           ... sto(p)
```

If no reply returns we should search for the error in the steps carried out so far.

Setr connected in such a way is only able to communicate with the local CU. Connection with other MORSE network CUs is possible after configuring MAS, see below.

**Note**

In the PC (WindowsXP especially) can be active the firewall, which prevents the ping receiving. Switch it off for the test purposes.

If we are working in LAN with the name server in which the CU address is defined we can enter the name of the computer instead of the IP address when starting Setr, e.g.:

```
setr -pIPradiomodem.racom.cz
setr -pIPradiomodem
```

## 4. Routing

Now we have to decide if we will use only one single MORSE address and one single application, or a group of 256 addresses and more applications (on more computers).

### 4A. Single address.

Add the connection to ETH channel in the Node menu:

```
Ne 1uE0   ... routing the user output from the node to the AAA app.
   1sE0   ... routing service packets to Setr
```

Set the EIe menu to the default condition:

```
EIe f     ... routing the user output of the Ethernet channel to node 1
```

## 5A. MAS configuration

Setting the EPe menu for MAS mode:

```
EPe 0t s1        ... link to Art table number 1
       B690F5600 ... Base - required PC MORSE address
       M00000000 ... zero mask requires an absolute match
                     for the Base and PC addresses
```

Now it is necessary to fill in Art table number 1. We have the possibility of doing this automatically or manually using Setr. For automatic filling switch off Setr (Alt+X) and then start it again using command:

```
setr -pIP10.0.0.2 -pm8000 -pw690F5600
```

where the meaning of parameters is:

```
-pIP10.0.0.2   ... IP address of CU
-pm8000        ... number of UDP port of application (Setr), in decimal
-pw690F5600    ... MORSE address, under which the application will be
                   known in the MORSE network, in hexadecimal
```

After starting Setr read the contents of the Art table:

```
ART No 1:
items: 1
default gw: 00000000 (0.0.0.0        )
dest:    gw:
690F5600 1F400001 (105.15.86.0     31.64.0.1      )
```

The Art table contains:

- in the `dest` column the MORSE address, under which application AAA is known in the MORSE network
- in the `gw` column there is a composite expression - let´s note that 0x1F40 is 8000 in decimal, which is the UDP port of our AAA application and furthermore 0x0001 is the lower part of the IP address of the computer running windows, from which only the lower 14 bits are valid and the upper 2 bits are zero.

We can fill the Art table in this manually for the necessary combination of addresses. It is useful to block automatic filling with the command:

```
EPe 0t ef    ... (e)nable:OFF
```

Appearance of the respective part of the menu:

```
MAS:
  (s)Art:1; write (e)nable:OFF
  (B)ase:690F5600  (M)ask:00000000
```

Now MAS is ready to transfer UDP datagrams from the AAA application, operating at the IP address 10.0.0.1 and at port 8000, which appears as address 690F5600 in the MORSE network.

We can check functionality using Setr. Switch Setr off and start with the parameters:

```
setr -pIP10.0.0.2 -pm8000 -pw690F5600        or
setr -pI0A000002 -pm0x1F40 -pw690F5600

>>!h
690F5600>sts
690F5600>1245  ... example of a reply
```

If no reply arrives search for the error in the steps according to points 4. and 5.

## 4B. More addresses, connecting Setr via LAN

Situation - The MR400 and several PCs are connected to the LAN network. Through the common MR400, on which MAS is running, it is possible to connect these PCs to the MORSE network. Each PC in the MORSE network is represented by a different MORSE address. Other applications may also be connected via the LAN network in this way.

Configuration:

- MR400 has for example the address 690F5600
- The PCs use addresses 690F5701, 690F5702, 690F5703, ....

In the `Ne` menu we set the wide table, which will route packets for the PC to the Link output. Switch on address multiplication, which sends packets originally routed to Link to the User output, and connect the User and Service output to E00:

```
  Nodes:
                                  retab
Nid|address |M | u   s | L   N |l w n g H|sTO Err  Cent vTO hTO
(0) 0048E62D     -  S00| -  R00|0 0 0 0 -| 15 SERV  OFF 304  30
(1) 690F5600 L  E00 E00| -  R01|0 1 0 0 -| 15 SERV  OFF 304  30
(2) 00000000    S01 S00| -  R02|0 0 0 0 -| 15 SERV  OFF 304  30
(3) 00000000    S02 S00| -  R03|0 0 0 0 -| 15 SERV  OFF 304  30
(4) 00000000    S03 S00| -  R04|0 0 0 0 -| 15 SERV  OFF 304  30
```

The routing wide table sends packets which have the PC destination address to the actual address of the node:

```
>>
Wide retab. No 1
57to:5600
```

The EIe menu remains in the default condition:

```
 Channel to Node Interface:
   retranslation     |   user+service             lim
 id N  A t          m | N  A t Base     m  sec brc S   e
(0) 0   NO AR         | 1   NO AR          usr OFF NONE
```

If we are working in the LAN network, the selection of the IP addresses of the application (PC with Setr) must also suit this network, for example:

```
IP adresa - 192.168.2.1
maska     - 255.255.255.0
```

In the EPe menu we select the IP address in accordance with the LAN network:

```
  Internet Protocol:
Eid| ip address            |    net mask            |          gw      |
(0) C0A80205 192.168.2.5     FFFFFF00 255.255.255.0   00000000 0.0.0.0
```

## 5B. MAS configuration

Choose `EPe Ots`. By selecting `(s)Art` in the parameters we activate MAS, `(B)ase` and `(M)ask` determine the area in which MORSE addresses of connected Setr´s can be selected:

```
MAS:
  (s)Art:1; write (e)nable:ON
  (B)ase:690F5700  (M)ask:000000FF
```

Start setr in windows using the command:

```
  setr -pIP192.168.2.5 -pm8000 -pw690F5701
```

where is:

```
  -pIP192.168.2.5 ... IP address of CU
  -pm8000         ... is the application port number, decimal
  -pw690F5701     ... is the MORSE address, under which the PC with Setr
                      will be known in the MORSE network
```

Other Setr may also be started at the same time on other PCs, however it is necessary that they all use a different MORSE address. After starting Setr we can look at how the Art table was automatically filled, here 2 Setr were started:

```
ART No 1:
items: 2
default gw: 00000000 (0.0.0.0        )
dest:    gw:
690F5702 1F40020C (105.15.87.2     31.64.2.12     )
690F5701 1F400201 (105.15.87.1     31.64.2.1      )
```

## 6. Packet transfer to the MORSE network

Routing, which ensures the sending of packets for the PC (here 690F57xx) to the MR400 address (here 690F5600), must be set in the destination CU. In the CU 690F5609 it is table wide:

```
690F5609h>
Wide retab. No 1
57to:5600
```

Connection example and the RFC monitoring in CU 690F5600:

```
690F5609h>!

690F5609h>

u E00   690F5600    R01
```

```
29/ 65   690F5609    serd

 serd    690F5609     R01
28/ 64   690F5600   u E00
690F5609h>

Monitoring: source 690F5600|1.
RF mon       |toa       frm      |dst       src      |lNo!DQ!RSS size|TT N
15:37:23.058|690F5609 690F5600|690F5609 690F5701|012  RFTX    10 98 1dat
15:37:23.105|690F5600 690F5609|                  |012*29~ 64    0*06  ack
15:37:23.155|690F5600 690F5609|690F5701 690F5609|012*28* 64   26*9A 1dat
15:37:23.155|690F5609 690F5600|                  |012  RFTX     0 06  ack
```

In the remote CU 690F5609 we can set the Async link protocol and direct it to the address of our application. The two lower bytes of IP application address, here 5701, are set in the protocol parameter. The 16 bit mask is set by the command `SIe 2um16`. Start monitoring of the port with Async link in the remote CU, route it to the address of our application.

Test the connection from Windows to the remote Async link. Use the function `mtU` for creating the necessary IP test frame. It sends the AAAA characters to the address choosen by command `!h690F5609`.

```
690F5609h>mtU

Send type :09
690F5609h>
07:53:31.393 tx      2 | S02
AAAA
```

Test the connection from the remote link to the application:

```
690F5609h>in                  ... use the function Channel data send
690F5609h>dc                  ... choose the data input in SCC2
690F5609h>ah0123456789ABCDEF  ... insert some data (here hex)
690F5609h>s                   ... start

690F5609h>O.K.
690F5609h>SETR: Packet type 09. 690F5701 690F5609  0889    8
0123 4567 89AB CDEF     ... application Setr received and printed the data

08:01:48.087 rxsim   8 | S02
0123 4567 89AB CDEF     ... monitoring input of data to the remote CU
```

**Note**

This is the connection with the Async. link in the remote CU. There may be a lot of CUs connected via MORSE network in this way and each can use a different protocol if necessary. Their data is then processed by the central application AAA connected through MAS, which distinguishes protocols best according to the respective MORSE addresses. This system integration is thus already contained in the MORSE system.

## UDP packet

If we want to view the format of the UDP packet we can start Setr with the parameter -mh and Setr will print data in the UDP datagrams. It is a good idea to switch off remote monitoring for this activity so that Setr does not monitor monitoring packets. On the contrary local monitoring in the CU, which contains MAS is switched on here. Again function `mtu`, which generates 4 characters AAAA in the Setr application is used:

```
690F5609h>mtU

Send type :09
07:39:18.437 tx 10 to:192.168.2.5, 8888
0000 8982 690F 5609 AAAA
```

... application Setr is started with parameter -mh. It sends a UDP datagram containing the MORSE pseudoframe 0000 8986 690F 5609 AAAA with destination MORSE address and data AAAA to IP address 192.168.2.5 and to port 8888.

```
08:39:30.663 rsi:52 rx|0002A94B108E |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A94B 108E 00C0 9F63 CFAB 0800 4500 0026 00F2 0000 8011 B47E
C0A8 0201 C0A8 0205 1F40 22B8 0012 4535 0000 8982 690F 5609 AAAA
```

... monitoring of the datagram entering the Ethernet channel contains, amongst others:

```
source IP address       C0A8 0201
destination IP address  C0A8 0205
source port             1F40 hex = 8000 dec
destination port        22B8 hex = 8888 dec
MORSE pseudoframe       0000 8982 690F 5609 AAAA

RF mon      |toa      frm     |dst      src     |lNo!DQ!RSS size|TT N
08:39:30.664|690F5609 690F5600|690F5609 690F5701|036  RFTX     2 89 2dat
AAAA
```

... monitoring of data sent through the MORSE radio channel

### Monitoring through MAS

When routing monitoring from the remote CU to the Setr connected through MAS it is necessary to insert the MORSE address of the PC with Setr into the `ise` menu. In the above example of the single application (A) it was address 690F5600, in the case of (B) it is, for example, address 690F5701. Use option `L` to insert it into the remote CU:

```
690F5609>
ise 0N1
    0aL     (type 'L' for local address)
```

The resulting menu ise contains the PC destination address (MORSE):

```
690F5609h>
System channels:
(Service 'iMo' works for s0 and s1 only)
id|--Node--addr------timeout---size---s(e)c--
(0)  1    690F5701     888     400     ON
```

```
(1)  0   00000000   888   400   ON
(2)  0   00000000   888   400   ON
(3)  0   00000000   888   400   ON
(4)  0   00000000   888   400   ON
(5)  0   00000000   888   400   ON
```

We then route monitoring to system channel 0.

## 7. Running the application

Using Setr switch off remote and local monitoring, switch off Setr, start application AAA and now only observe how everything operates. The application runs on UDP port 8000 and communicates against UDP port 8888, which is used by the CU. The format of data is described in another document (Format of UDP datagram IPGW for MORSE).

## SUMMARY

Brief summary of the steps described above for connecting the application through the MORSE Application Server (MAS):

1. IP address in PC:

```
IP address - 10.0.0.1
mask       - 255.255.255.0
```

IP and MORSE address in CU:

```
EPe 0i0A000002    ... i.e. 10.0.0.2
    0nFFFFFF00    ... i.e. 255.255.255.0
Ne  1a690F5600    ... MORSE address
```

2. Ping from PC to CU:

```
ping 10.0.0.2
```

3. 

```
setr -pIP10.0.0.2
```

ping from CU to PC:

```
EPe 0tP t0A000001   s e p
```

4A. Connection through MAS with one address:

```
Ne 1uE0
   1sE0
EIe f
```

5A. Setting MAS:

```
EPe 0t s1
       B690F5600
```

```
       M00000000

setr -pIP10.0.0.2 -pm8000 -pw690F5600
```

Content of Art:

```
dest:    gw:
690F5600 1F400001 (105.15.86.0     31.64.0.1 )
```

4B. More addresses, connection of Setr through LAN:

```
Ne 1a690F5600
   1MLn
   1uE0
   1sE0
   1w1
Tw1
57to:5600
EIef
EPe 0 iC0A80205
      nFFFFFF00
```

5B. Setting MAS:

```
EPe 0 t
       s1
       B690F5700
       M000000FF

setr -pIP192.168.2.5 -pm8000 -pw690F5701
```

Content of Art:

```
dest:    gw:
690F5702 1F40020C (105.15.87.2     31.64.2.12  )
690F5701 1F400201 (105.15.87.1     31.64.2.1   )
```

6. Packet transfer through the net:

destination CU 690F5609:

```
Tw1
57to:5600

u E00   690F5600    R01
30/ 70  690F5609   serd

 serd   690F5609    R01
29/ 68  690F5600  u E00
```

Test connection from Setr through MAS to the remote async link:

```
690F5609h>mtU

Send type :09
690F5609h>
07:53:31.393 tx      2 | S02
AAAA
```

# 6. Format of UDP datagram IPGW for Morse

This datagram is used above all for Morse Application Server (MAS) and for internal use in Racom for IP Retranslation of Morse packets (M-IP-M). IPGW = Internet Protocol Gate Way.

The UDP datagram wraped in the IP-datagram has this structure (including the example):

```
| ETH header | IP header | UDP header | MORSE pseudoframe |

08:39:30.663 rsi:52 rx|0002A94B108E |00C09F63CFAB | IP/UDP/MOR/USR/DATA
0002 A94B 108E 00C0 9F63 CFAB 0800 4500 0026 00F2 0000 8011 B47E
C0A8 0201 C0A8 0205 1F40 22B8 0012 4535 0000 8982 690F 5609 AAAA
```

ETH, IP and UDP header are described in the literature, they are mentioned here only for better orientation. The main part MAS follows this paragraph.

The length of all IPGW frame is written in the monitoring header, here 52 bytes.

### ETH header

```
|                 ETH header               |
|   dst eth /48   |  src eth /48  | prot/16 |
   0002 A94B 108E   00C0 9F63 CFAB    0800
```

Meaning of items:

| | | |
|---|---|---|
| dst eth /48 | dst ETH - destination address | |
| src eth /48 | src ETH - source address | |
| prot/16 | 0800 | IP-datagram follows |
| | 0806 | ARP follows |

### IP header

```
|                             IP header                                   |
|vers/16|IPhlen/16| No/16 |frag/16|par1/16|hchs/16| src IP/32 | dst IP/32|
   4500    0026     00F2    0000    8011    B47E    C0A8 0201   C0A8 0205
```

Meaning of items:

| | | |
|---|---|---|
| vers/16 | ver/4 | version of IP protocol (now 4) |
| | headlen/4 | length of IP header (here 5 words by 32 bits) |
| | type serv/8 | type of desired service for datagram transmission |
| IPhlen/16 | number of bytes in IPGW frame without ETH header, hex (here 26 hex = 38 dec) | |
| No/16 | order number, datagram identification, hex | |
| frag/16 | fragmentation: | |
| | res/1 | reserve |
| | no/1 | 1=datagram fragmentation forbidden |
| | next/1 | 1=next fragments follow |

| | | |
|---|---|---|
| | offset/13 | fragment offset- position of beginning of fragment data part in regard of underived datagram(in bytes) |
| par1/16 | time to live/8 | lifetime of datagram in seconds, by passing through router it is decremented at least by 1, at =0 it is discarded |
| | protocol/8 | determines the protocol of higher level, which report is contended in data part of datagram |
| hchs/16 | checksum including IP head only | |
| src IP/32 | src IP - source address, see menu EPe 0i = (i)p addr | |
| dst IP/32 | dst IP - destination address, see menu EPe 0i | |

### UDP header

```
|               UDP header                     |
| src port/16 | dst port/16 | UDPlen/16 | chs/16 |
     1F40          22B8         0012      4535
```

Meaning of items:

| | |
|---|---|
| src port/16 | source port UDP |
| dst port/16 | destin port UDP ( 22B8 hex = 8888 dec ) |
| UDPlen/16 | length of UDP packet in bytes(UDP header + pseudoframe),hex (here 12 hex = 18 dec) |
| chs/16 | checksum |

### MORSE pseudoframe

According to first bit of pseudoframe is IPGW frame divided into 2 groups:

- A - first bit is zero - Morse Application Server
- B - first bit is one - proprietary for Racom, UDP frames for M-IP-M mode

### Pseudoframe (A) - Morse Application Server

```
|                MORSE pseudoframe                 |
|flags/16|PT/8|D/1|  R/4 | No/3| addr/32 |  data  |
|  0000  | 89 | 1 | 0000 | 010 |690F 5609|  AAAA  |
```

Meaning of items:

| | | | |
|---|---|---|---|
| flags/16 | T/4 | 0x0 = 0000 | UDP datagram type MAS |
| | R/4 | 0x0 = 0000 | reserve |
| | U/8 | subtype | 0x00 - user data |
| | | | 0x01 - seek/delete format, for internal use in Racom only |
| PT/8 | MORSE packet type | | |
| D/1 | DTE bit | 0- DCE sending=CU in MORSE net | |

| | | |
|---|---|---|
| | 1- DTE sending=IP application | this part is coincidently |
| R/4 | reserve (0000) | similar like net level packet |
| No/3 | packet order number | of MARS-A protocol |
| addr/32 | CU address in MORSE net | |
| data | data transmitted, length (UDPlen-0x10) byte | |

## APPENDIX:

MORSE pseudoframe B), for internal use only. These datagrams are used for M-IP-M communication i.e. from MORSE network through IP network and to the MORSE again.

| | |
|---|---|
| B1 - UDP data frames and fragments | 0xD,0xB,0xF,0xE |
| B2 - UDP appended frame | 0xA |
| B3 - UDP control frame | 0xC |

### Pseudoframe (B1) - UDP data frames and fragments

```
|                        MORSE pseudoframe
| flags/16 | No/16 | to/32   |  from/32 |type/16| dest/32 |  src/32 |
    D200     1392   690F8909   690F8101   0B89   690F8909   690F8101

      MORSE pseudoframe          |
| [mob/32] |    data    | crc/16 |
  ........      AAAA       2788
```

Meaning of items:

| | | | |
|---|---|---|---|
| flags/16 | T/4 | 0xD | data frame |
| | | 0xB | fragments begin |
| | | 0xF | fragment |
| | | 0xE | fragments end |
| | A/1 | appended frame | |
| | r/1 | repeated bit (1 if the packet is repeated) | |
| | S/1 | security bit (1 requests ACK) | |
| | P/1 | transmitter problem bit | |
| | R/4 | reserve | |
| | pver/4 | protocol version | |
| No/16 | Link No | | |
| to/32 | to address | | |
| from/32 | from address | | |
| type/16 | packet full type | | |
| dest/32 | destin. address | | |
| src/32 | source address | | |

| | |
|---|---|
| mob/32 | mobile address, for mobile service only, i.e. if (type & 0xE000) is nonzero |
| data | data transmitted, length (UDPlen-0x20) byte |
| crc/16 | Cyclic Redundancy Check |

### Pseudoframe (B2) - UDP appended frame

```
|                  MORSE pseudoframe                    |
| flags/16 | res/5 | lenA/11 | No/16 |   data  | crc/16 |
    AA00         0004          0005     AAAA     ....
```

Meaning of items:

| | | |
|---|---|---|
| flags/16 | T/4 | 0xA appended frame |
| | A/1 | appended frame |
| | r/1 | repeated bit (1 if the packet is repeated) |
| | S/1 | security bit (1 requests ACK) |
| | P/1 | transmitter problem bit |
| | R/1 | reserve |
| | pver/4 | protocol version |
| res/5 | reserve | |
| lenA/11 | length (No + data), byte | |
| No/16 | Link No | |
| data | data transmitted, length (UDPlen-0x10) byte | |
| crc/16 | Cyclic Redundancy Check | |

### Pseudoframe (B3) - UDP control frame

```
|               MORSE pseudoframe                     |
| flags/16 | CN/16 | Cf/16 | frm/32 | toa/32 | crc/16 |
   C100      1392    D200   690F8101 690F8909   EE98
```

Meaning of items:

| | | |
|---|---|---|
| flags/16 | T/4 | 0xC control frame |
| | A/1 | appended frame (=0) |
| | R/1 | reserved (=0) |
| | CT/2 | control type |
| | | 00 NONE |
| | | 01 ACK |
| | | 10 REJ |
| | | 11 NAK |
| | R/4 | reserved |
| | pver/4 | protocol version |
| CN/16 | control link No (link No of confirmed frame) | |
| Cf/16 | control flags (flags of confirmed frame) | |

| | |
|---|---|
| frm/32 | from address |
| toa/32 | to address |
| crc/16 | Cyclic Redundancy Check |

Used examples for appendix (B1,B3):

```
14:30:20.090 rsi:68 tx|0002A95AA517 |0002A94AE97E | IP/UDP/MOR/RET/DAT
0002 A95A A517 0002 A94A E97E 0800 4500 0036 0073 4000 4011 88E9 C0A8 1001
C0A8 2009 22B8 22B8 0022 8D3C D200 1392 690F 8909 690F 8101 0B89 690F 8909
690F 8101 AAAA 2788
14:30:20.090 rsi:58 rx|0002A94AE97E |0002A95AA517 | IP/UDP/MOR/RET/CTL/ACK
0002 A94A E97E 0002 A95A A517 0800 4500 002C 000C 4000 4011 895A C0A8 2009
C0A8 1001 22B8 22B8 0018 979C C100 1392 D200 690F 8101 690F 8909 EE98
```