The perspectives of the Amateur University Networks – AMUNETs (Part 3)

MIROSLAV SKORIC
Faculty of Technical Sciences
University of Novi Sad
Skoric, P. O. Box 52, 21102 Novi Sad
SERBIA
skoric@uns.ac.rs http://tldp.org/HOWTO/FBB.html

Abstract: - This paper discusses results of recent experiments, which intention was to simulate direct and indirect access to a remote amateur radio e-mail server from a chain of amateur radio relay systems (nodes). We based the experiments on publicly available node-networking software, which is compatible with Microsoft Windows and MS DOS / PC DOS operating systems. Nevertheless, a prospective experimenter is also encouraged to perform tests by using Linux operating system and appropriate solutions that come with that OS. In addition to the opportunities for experimenting with repeater programs for personal computers, networking procedures and commands are implemented in non-computer, or so-called 'hardware' types of radio relay devices. This work suggests the readers a couple of basic methods of communication in a wired local area network (LAN) which aim is to give a picture of what the amateur radio networking is about and how it can be practically implemented within an educational computer network.

Key-Words: - amateur radio, AMUNET, communication, computer, relay, node, university, education

1 Introduction

Amateur radio gives us so many communicating opportunities to explore. Some amateur radio enthusiasts tend to establish wireless links with their peers by manual Morse telegraphy, while the others use their computers. As described in [1], the preferred method of establishing computer-related communications at relatively short distances (varying from several dozens of meters up to several hundreds of kilometers – in case of a flat terrain) is a direct link between two or more 'packet-radio' correspondents, as shown in Figure 1.

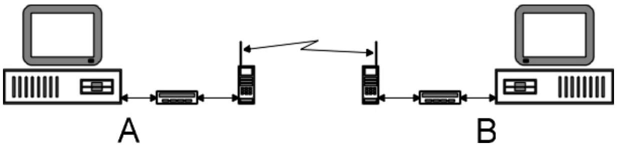


Fig.1. Direct communication

The best practice is if two correspondents negotiate the working frequency and switch their transmitters to an unoccupied channel, commonly named as 'simplex' channel, in order to continue their conversation. Unfortunately, using such a simple mode is fine only if there are no natural or artificial obstacles between the two, which often prevent radio signals to go thru. In opposite, the usage of co called amateur radio 'digital repeaters' (shortly 'digipeaters'), or 'nodes', which are usually constructed and maintained by various amateur radio groups, unions and societies that exist all around the world – particularly in large metropolitan areas, is unavoidable. A packet-radio node ('R' in Fig. 2) is a relatively simple combination of a radio station and a modem, placed on a building or on a mountaintop.

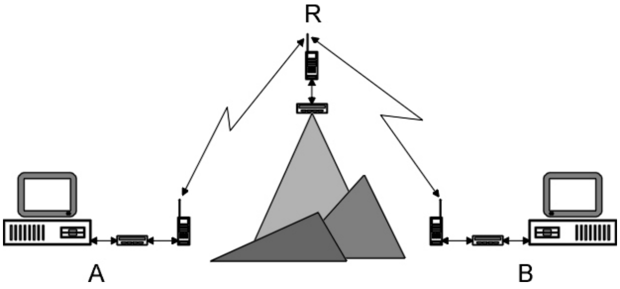


Fig.2. Indirect communication via radio-relays

As you can see, we depicted the radio station and the modem at 'R' to look like as if they are the same or similar type as those used by the end-users 'A' and 'B'. Actually, we did it intentionally because it is often the case, particularly when a *packeteer* brings along his or her equipment to a mountain or some other countryside to spend a weekend or vacation and exercise the role of a repeater. Nevertheless, despite their visible similarity, there are some differences. In fact, the modem design at the end-users' side might be even more simplified because some of their functionalities in handling digital data may be transferred to the computer's central processing unit (CPU) and/or communication programs the end-users have been using in their communicating activities. In opposite to that, the relay systems usually do not have computers nearby, so they cannot take advantages of the computers' CPUs or so. That is the reason why many 'hardware' relay systems are constructed by using modems of so-called TNC2 (abbr. terminal-node-controller-type-two) category. TNC2 devices are equipped with internal CPUs, random access memory (RAM) and read-only memory (ROM) [2]. While temporary data are stored in RAM, the relay programs reside in ROM. Some of the programs for TNC2-based amateur radio relay systems are The Net 2.06 and up, as well as X1J. One of the drawbacks of those programs is that their functionalities are limited to available storage area within the modem's ROM, which is usually 256 kilobytes. Another issue is the modem's RAM, whose capacity is similar to the ROM, and that low capacity does not allow any large amount of working data to be stored for a long time. Both limitations mentioned become an issue – soon or later and many amateur radio groups have decided to switch from those 'hardware' repeaters to the 'software' ones. The latter run as computer programs in IBM-compatible personal computers. During the last two centuries, the radio amateurs authored various software flavors for almost all popular operating systems, including MS DOS / PC DOS, MS Windows and Linux. Some of the advantages of implementing computer-based relay systems are the following: Large hard disks that are capable to store practically indefinite amounts of data, speedier CPU's of modern PC's, which handle complex data relaying operations, etcetera.

In the next section, we are going to test interaction between two software-based radio relay solutions. As a starting consideration, we will assume that digital repeater 'R1' (see Figure 3) is a network entry point for our end-user 'A'. Let us suppose that 'R1' is a stand-alone device mounted on a nearest mountaintop. One of its neighboring nodes, digital repeater 'R3', runs from a university premises (for example from a building's top floor). In addition, 'R3' has a wired link to a computer in a school's lab that serves as an amateur bulletin board system (BBS), i.e. an e-mail server.

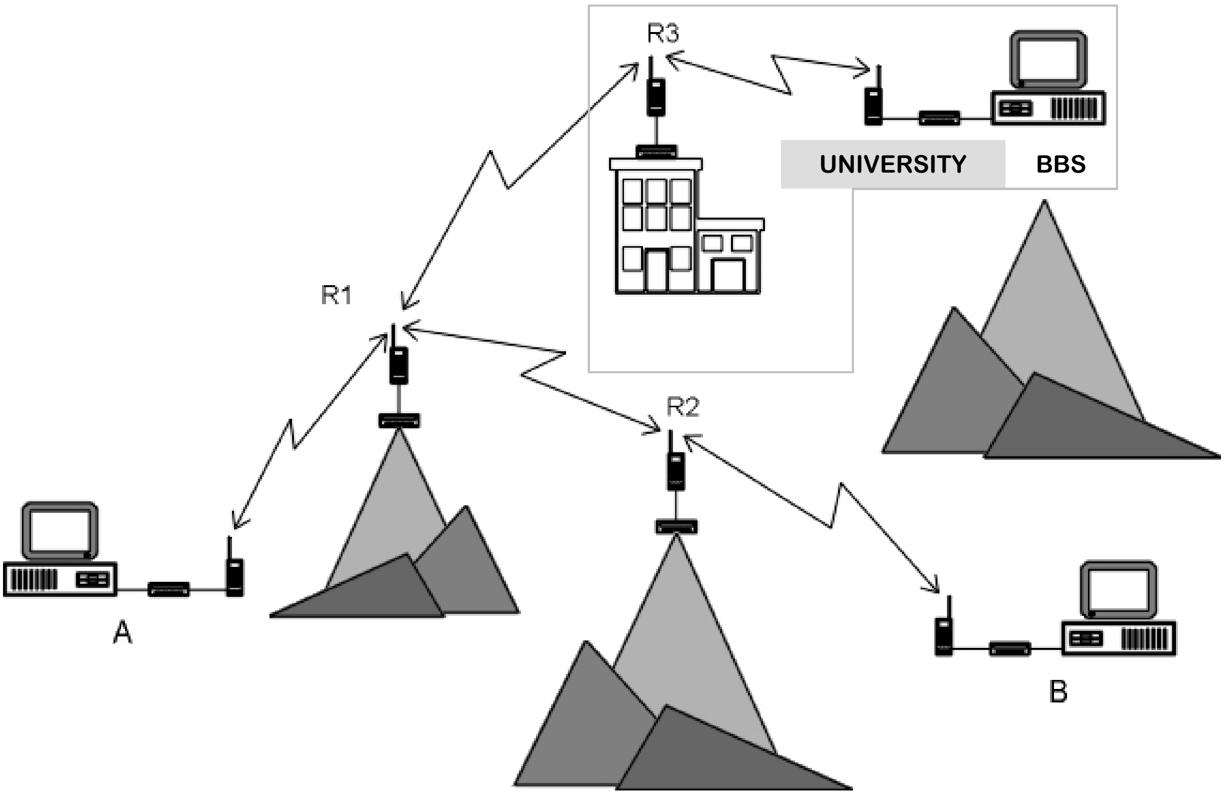


Fig.3. An educational AMUNET interconnects a university and surrounding schools (A, B)

2 Experimentation

2.1 Purpose

The intention of our experiments is to simulate a real amateur radio network that consists of at least two relay systems, 'R1' and 'R3' and examine its behavior. The node 'R1' is supposed to serve a role of the entry point to the radio amateurs who live in its area of coverage (which can be from some dozens of meters to more than 20 kilometers around it). Having that in mind, 'R1' has to be accessible to the local users, including the user 'A'. In our example, that means the first node has an amateur radio frequency input/output at 144.875 MHz, working at the standardized speed of 1200 bits per second. The other planned relay system, the node 'R3', will be located at the nearest university, for example at the top floor of the main academic building. Besides that, there is an email server, or as it is called in the amateur radio terminology – a BBS (Bulletin Board System).

The working plan for a user 'A' is to establish communication with an existing packet radio network, by using his or her nearest node (in this case, it is 'R1'). From that point, the user 'A' will continue to the node 'R3' and from there to login to the email server to access his or her mailbox. The purpose of the experiments is to show available procedures to prospective amateur radio newcomers.

2.2 Apparatus

We used the following equipment for our experiments:

Node 'R1'
- Computer PC AT 80286 CPU clock 12 MHz, 1 MB RAM
- Operating system MS DOS 5.0
- Network node software BPQ 4.08a for MS DOS / PC DOS
- Modem Symek TNC2S
- VHF radio station Icom IC-2GE

Node 'R3'
- Computer P2 Celeron CPU clock 400 MHz, 224 MB RAM,
- Operating systems MS Windows XP, MS Windows 2000
- Network node software BPQ 4.10f for Windows
- E-mail server software WinFBB 7.00i for Windows

Technical elements of 'R1' make it fully capable to communicate with the other radio amateurs, i.e. all packet-radio stations its antenna can 'hear'. In order to simulate radio communication with the other node(s), we equipped our computer (Figure 4, middle) with a network card that made it possible to communicate with the other computer(s).

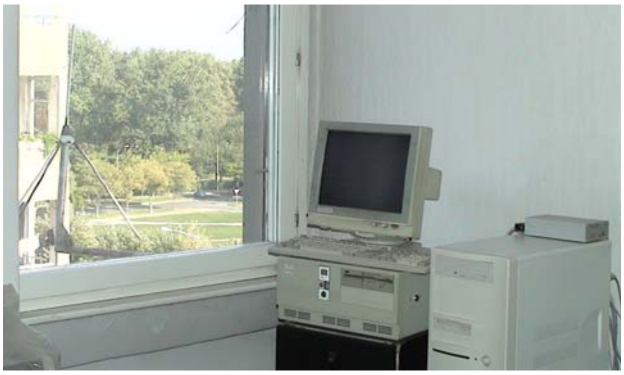


Fig.4. Node 'R1' elements: Outdoor antenna (left), control computer (middle), modem (top right), VHF radio station (behind the computer)

The 'R1' modem, shown in Figure 5, is set to so-called KISS (= 'Keep It Simple, Stupid') mode with rudimentary internal commands, to be compatible with the complexity of the BPQ program:

Fig.5. The modem of 'R1' has two *firmware* (software installed within its EPROM). We configured one of them, *TAPR 1.1.8*, to work in KISS mode

On the other side of our simulated link is the virtual node 'R3'. It does not have any physical radio frequency input/output and instead of that, it is equipped with another network interface card for a 10 MB per second connection to 'R1' (Figure 6). In addition, on the same computer, we have installed the popular email-server software *WinFBB* [3] of version 7.00i. We configured our email-server to accept incoming connection requests that come through the BPQ32 node in the same manner, as it would do by using the real radio channels.

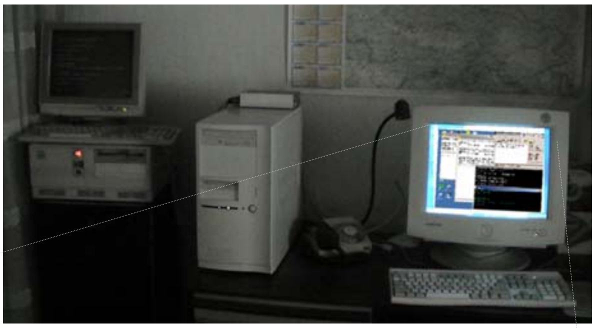
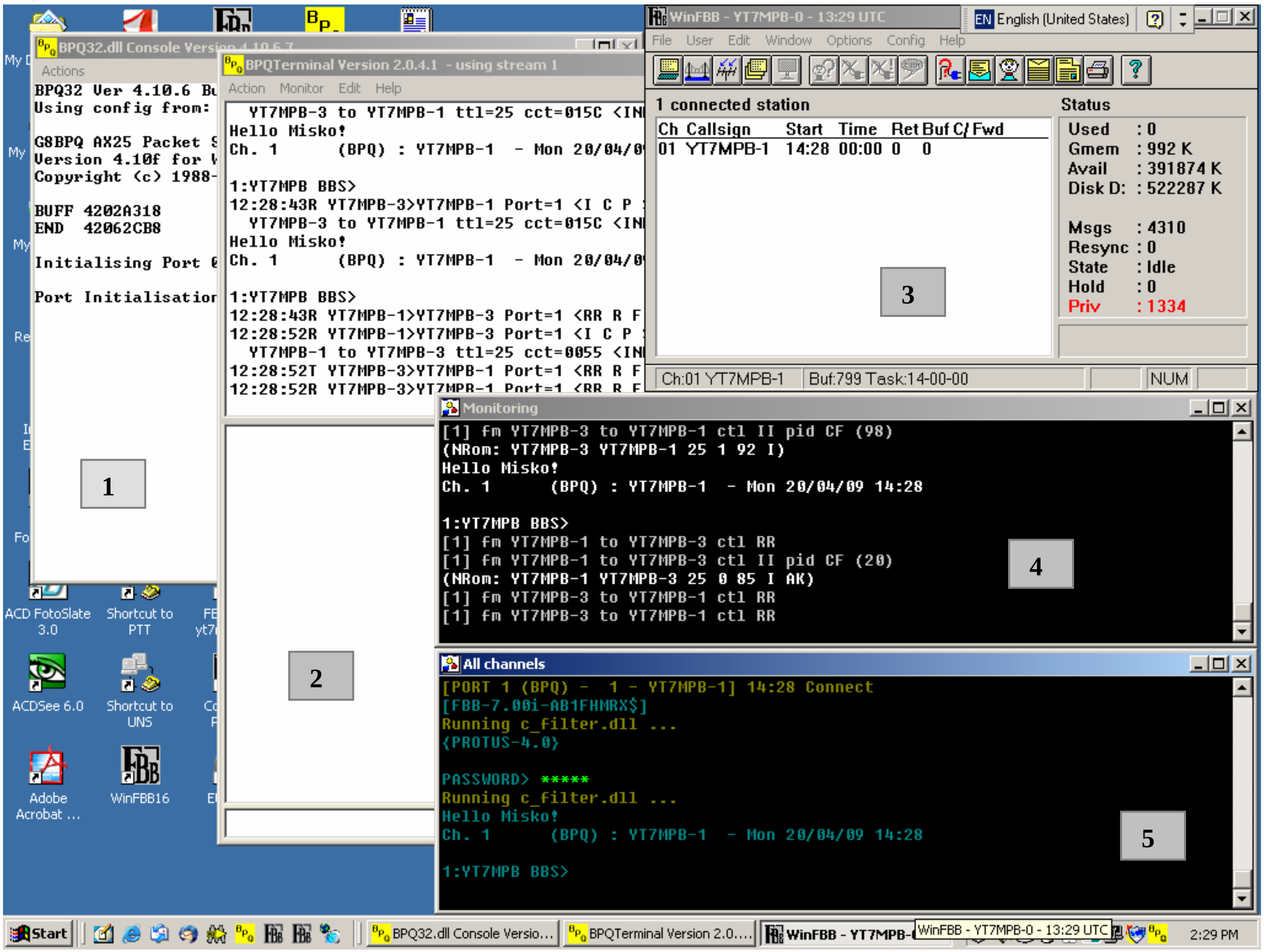


Fig.6. Experimental test bed: Node 'R1' (far left), 'R3' control + BBS e-mail server (middle and right)



BPQ32.dll Console Version 4.10.6.7

```
BPQ32 Ver 4.10.6 Bu
Using config from:

G8BPQ AX25 Packet S
Version 4.10f for W
Copyright (c) 1988-

BUFF 4202A318
END  42062CB8

Initialising Port 0

Port Initialisation
```

BPQTerminal Version 2.0.4.1 - using stream 1

```
  YT7MPB-3 to YT7MPB-1 ttl=25 cct=015C <IN
Hello Misko!
Ch. 1       (BPQ) : YT7MPB-1  - Mon 20/04/0

1:YT7MPB BBS>
12:28:43R YT7MPB-3>YT7MPB-1 Port=1 <I C P
  YT7MPB-3 to YT7MPB-1 ttl=25 cct=015C <IN
Hello Misko!
Ch. 1       (BPQ) : YT7MPB-1  - Mon 20/04/0

1:YT7MPB BBS>
12:28:43R YT7MPB-1>YT7MPB-3 Port=1 <RR R F
12:28:52R YT7MPB-1>YT7MPB-3 Port=1 <I C P
  YT7MPB-1 to YT7MPB-3 ttl=25 cct=0055 <IN
12:28:52T YT7MPB-3>YT7MPB-1 Port=1 <RR R F
12:28:52R YT7MPB-3>YT7MPB-1 Port=1 <RR R F
```

WinFBB - YT7MPB-0 - 13:29 UTC

```
1 connected station                          Status
Ch Callsign    Start  Time  Ret Buf C/ Fwd   Used    : 0
01 YT7MPB-1    14:28  00:00 0   0            Gmem    : 992 K
                                             Avail   : 391874 K
                                             Disk D: : 522287 K

                                             Msgs    : 4310
                                             Resync  : 0
                                             State   : Idle
                                             Hold    : 0
                                             Priv    : 1334
Ch:01 YT7MPB-1   Buf:799 Task:14-00-00   NUM
```

Monitoring

```
[1] fm YT7MPB-3 to YT7MPB-1 ctl II pid CF (98)
(NRom: YT7MPB-3 YT7MPB-1 25 1 92 I)
Hello Misko!
Ch. 1       (BPQ) : YT7MPB-1  - Mon 20/04/09 14:28

1:YT7MPB BBS>
[1] fm YT7MPB-1 to YT7MPB-3 ctl RR
[1] fm YT7MPB-1 to YT7MPB-3 ctl II pid CF (20)
(NRom: YT7MPB-1 YT7MPB-3 25 0 85 I AK)
[1] fm YT7MPB-3 to YT7MPB-1 ctl RR
[1] fm YT7MPB-3 to YT7MPB-1 ctl RR
```

All channels

```
[PORT 1 (BPQ) -  1 - YT7MPB-1] 14:28 Connect
[FBB-7.00i-AB1FHMRX$]
Running c_filter.dll ...
{PROTUS-4.0}

PASSWORD> *****
Running c_filter.dll ...
Hello Misko!
Ch. 1       (BPQ) : YT7MPB-1  - Mon 20/04/09 14:28

1:YT7MPB BBS>
```

1
2
3
4
5
Actions
Action Monitor Edit Help
File User Edit Window Options Config Help
EN English (United States)
ACD FotoSlate 3.0
Shortcut to PTT
ACDSee 6.0
Shortcut to UNS
Adobe Acrobat ...
WinFBB16
Start
BPQ32.dll Console Versio...
BPQTerminal Version 2.0....
WinFBB - YT7MPB-0 - 13:29 UTC
2:29 PM

Fig.7. Node 'R3' subsystem (#1 and #2); BBS subsystem (#3, #4, and #5)

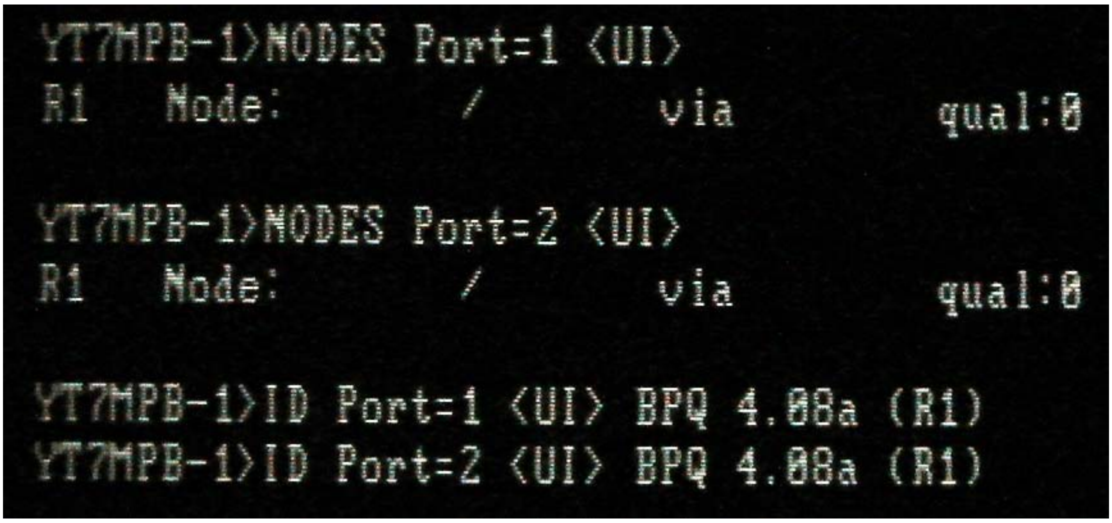
To explore functionalities of both sides in the 'R1'–'R3' link, it is needed to work step-by-step. During the first phase, we are going to activate the software at 'R1' computer, check the initial content of its working memory and leave it running alone for a while (5-10 minutes). During the second phase, we will activate: a) software running the 'R3' node, b) the BBS program on the same computer, control the memory content of 'R3' and then leave both nodes to run simultaneously. During the next phases, we will monitor both nodes' self-adaptive reconfigurations and perform some tests with accessing the e-mail server subsystem. Figure 7 shows all active programs at the side of node 'R3'. The functions of the opened windows are:

#1: To confirm that the node program is running,
#2: To administer the traffic that goes via the node,
#3: To control main functions of the e-mail server,
#4: To monitor the traffic on the server's outputs,
#5: To monitor activities of connected users and the traffic between the server and the users.

2.3 Test results

2.3.1 Phase 1

Upon its activation, the memory content of an amateur radio repeater should be empty, provided the repeater software has not started by reading some 'last-known-state' data, which some computer programs allow as a configurable option. In the case of 'R1', we did not use such an opportunity. Soon after we activated 'R1', it broadcasted its first 'nodes' list (as shown in Figure 8, the list was empty) and the additional information text *BPQ 4.08a (R1)* to announce its presence and identify itself. Note that 'R1' has two physical ports: 'Port=1', which is the radio port, and 'Port=2', which is the LAN interface, so the R1's broadcasts went out via both ports:



```
YT7MPB-1>NODES Port=1 <UI>
R1   Node:        /        via         qual:0

YT7MPB-1>NODES Port=2 <UI>
R1   Node:        /        via         qual:0

YT7MPB-1>ID Port=1 <UI> BPQ 4.08a (R1)
YT7MPB-1>ID Port=2 <UI> BPQ 4.08a (R1)
```

Fig.8. Repeater 'R1' in function

Because of the empty memory of 'R1', the repeater's internal commands ***n*** ('nodes' – which stands for virtual links to available nodes in a wider network) and ***r*** ('routes' – which stands for physical links to the neighboring nodes) did not return anything but the command prompt:

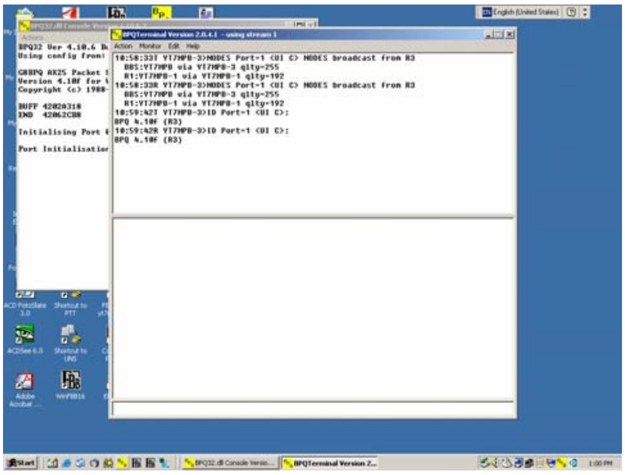
R1:YT7MPB-1} Nodes:
(a)
R1:YT7MPB-1} Routes:

2.3.2 Phase 2

In the case of 'R3', its ***r*** list ('routes') is also empty, though its ***n*** list ('nodes') has a single entry, *BBS:YT7MPB*, which is the BBS, an e-mail server that is 'hardwired' to the second node:

R3:YT7MPB-3} Nodes:
BBS:YT7MPB
(b)
R3:YT7MPB-3} Routes:

That is the situation at the 'cold' start of the nodes. Figure 9 describes the opened program screens of the second repeater, 'R3', after its software's activation and a short period of its work.



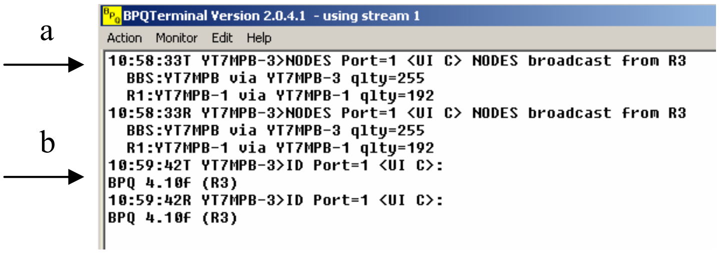
```
BPQ32 Ver 4.10.6 D
Using config from:

G8BPQ AX25 Packet S
Version 4.10f for W
Copyright (c) 1988-

BUFF 4202A318
END  42062C88

Initialising Port 0

Port Initialisation
```

```
10:58:33T YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
10:58:33R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
10:59:42T YT7MPB-3>ID Port=1 <UI C>:
BPQ 4.10F (R3)
10:59:42R YT7MPB-3>ID Port=1 <UI C>:
BPQ 4.10F (R3)
```

Fig.9. Repeater 'R3' in function

2.3.3 Phase 3

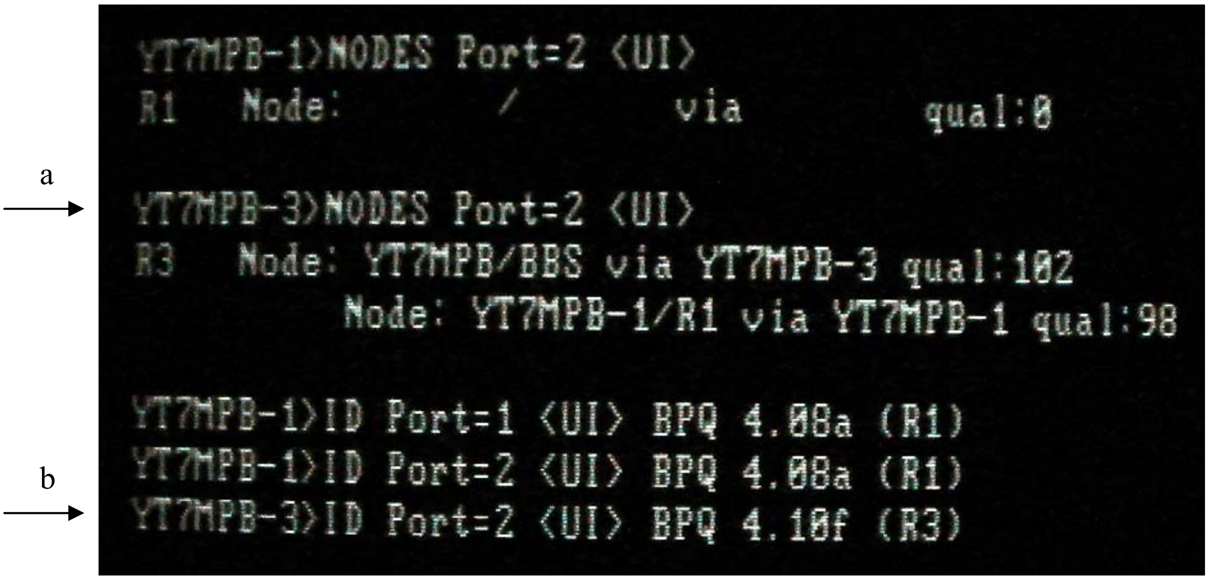
After a few minutes of apparent inactivity but according to its predefined schedule, 'R3' publicly broadcasts its 'table of nodes', which consists of: a) its own alias/callsign pair and the accompanying BBS's alias/callsign pair, as well as b) an additional information to identify its existence (Figures 10 and 11). Note that our virtual node 'R3' is equipped with only one physical port (Port=1, which is the LAN port), so its broadcasts went out via single port. However, the R3' broadcast appears twice in Figure 10, because the first occurrence is related to the actual transmission (T) while the second occurrence is related to an immediate reception (R) of what has been transmitted on the same port:



```
10:58:33T YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
10:58:33R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
10:59:42T YT7MPB-3>ID Port=1 <UI C>:
BPQ 4.10f (R3)
10:59:42R YT7MPB-3>ID Port=1 <UI C>:
BPQ 4.10f (R3)
```

Fig.10. The content broadcasted by 'R3'

The nodes send and receive their data by using predefined physical outputs, which can be the radio frequencies, Internet 'tunnels' – if they are available, or the local area interfaces as in our simulations. As described in Figure 11, 'R1' received on its Port=2 the information broadcasted by 'R3' (those are the middle paragraph starting with *YT7MPB-3>* and the last line, also starting with *YT7MPB-3>*). At that point, the 'R1' node immediately adapts to the incoming information, so that its own 'nodes' and 'routes' lists change. Although that change is fast, it is not visible at the first sight. Practically speaking, until a user invokes appropriate commands, we have been discussing about in this work, he or she will not know that those changes have even occurred.



```
YT7MPB-1>NODES Port=2 <UI>
R1   Node:       /          via          qual:0

YT7MPB-3>NODES Port=2 <UI>
R3   Node: YT7MPB/BBS via YT7MPB-3 qual:102
           Node: YT7MPB-1/R1 via YT7MPB-1 qual:98

YT7MPB-1>ID Port=1 <UI> BPQ 4.08a (R1)
YT7MPB-1>ID Port=2 <UI> BPQ 4.08a (R1)
YT7MPB-3>ID Port=2 <UI> BPQ 4.10f (R3)
```

Fig.11. 'R1' received the content broadcasted by 'R3'

Now, if we issue ***n*** and ***r*** commands at 'R1' again, we will get new responses:

```
R1:YT7MPB-1} Nodes:
BBS:YT7MPB      R3:YT7MPB-3
                                        (c)
R1:YT7MPB-1} Routes:
  2 YT7MPB-3 192 2
```

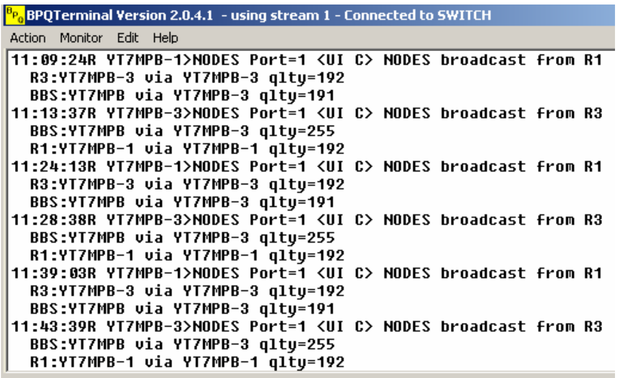
Let us discuss the changes at 'R1'. After comparing responses (a) and (c), it becomes obvious that the **n**(odes) list of 'R1' got populated with two new entries, *BBS:YT7MPB* and *R3:YT7MPB-3* while the **r**(outes) list of 'R1' got populated with one new entry, *2 YT7MPB-3 192 2*. The two new entries in the node list inform the visitors of 'R1' about the availability of those new nodes, which they can approach by using the command: ***c***(onnect) <alias/callsign>. The single new entry in the route lists describes the actual physical path to available nodes: the callsign of the other repeater, quality of the signal heard (*192*, which is very good), and the number of the R1's working port (*2*, which is the network card).

2.3.4 Phase 4

After a few minutes, 'R1' broadcasts its own table of nodes, which includes not only its own alias/callsign pair but also the new data it received from 'R3'. As expected, the 'R3' node receives that changed broadcast from 'R1', adopts its own memory content and almost immediately, the answers to ***n*** and ***r*** commands at 'R3' change to the following:

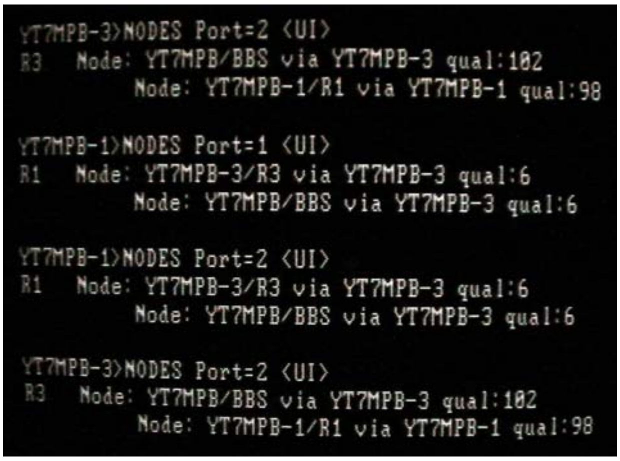
```
R3:YT7MPB-3} Nodes:
BBS:YT7MPB       R1:YT7MPB-1
                                          (d)
R3:YT7MPB-3} Routes:
  1 YT7MPB-1 192 1
```

Now if we leave both nodes alone – without any operator's intervention, in some half an hour they will exchange their tables of nodes repeatedly (see Figures 12 and 13) and as a result their 'nodes' list will remain almost identical – as long as the nodes are 'alive', i.e. as long as they continue with exchanging their information.


BPQTerminal Version 2.0.4.1 - using stream 1 - Connected to SWITCH
Action Monitor Edit Help

```
11:09:24R YT7MPB-1>NODES Port=1 <UI C> NODES broadcast from R1
  R3:YT7MPB-3 via YT7MPB-3 qlty=192
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:13:37R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
11:24:13R YT7MPB-1>NODES Port=1 <UI C> NODES broadcast from R1
  R3:YT7MPB-3 via YT7MPB-3 qlty=192
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:28:38R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
11:39:03R YT7MPB-1>NODES Port=1 <UI C> NODES broadcast from R1
  R3:YT7MPB-3 via YT7MPB-3 qlty=192
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:43:39R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
```

Fig.12. Mutual exchange of content (shown at 'R3')



```
YT7MPB-3>NODES Port=2 <UI>
R3    Node: YT7MPB/BBS via YT7MPB-3 qual:102
            Node: YT7MPB-1/R1 via YT7MPB-1 qual:98

YT7MPB-1>NODES Port=1 <UI>
R1    Node: YT7MPB-3/R3 via YT7MPB-3 qual:6
            Node: YT7MPB/BBS via YT7MPB-3 qual:6

YT7MPB-1>NODES Port=2 <UI>
R1    Node: YT7MPB-3/R3 via YT7MPB-3 qual:6
            Node: YT7MPB/BBS via YT7MPB-3 qual:6

YT7MPB-3>NODES Port=2 <UI>
R3    Node: YT7MPB/BBS via YT7MPB-3 qual:102
            Node: YT7MPB-1/R1 via YT7MPB-1 qual:98
```

Fig.13. Mutual exchange of content (shown at 'R1')

Let us compare (b) and (d). It is obvious that exchange of table of nodes resulted in updating R3's 'nodes' and 'routes' lists: Now the 'nodes' list of 'R3', besides the existing *BBS:YT7MPB*, has got a new entry of *R1:YT7MPB-1* and the previously empty 'routes' list of 'R3' got populated with a new entry, *1 YT7MPB-1 192 1*. Similarly to the updated status of 'R1', now the local users of 'R3' know that nodes *BBS:YT7MPB* and *R1:YT7MPB-1* are directly available at repeater 'R3' by entering the command ***c***(onnect) <alias/callsign>.

2.4 Available procedures

2.4.1 Procedure 1

Let us suppose that we have entered our packet-radio network by using the node 'R1' and now we want to access the BBS from it. We have a couple of possibilities. The first one is to execute an outgoing connection request to the second node in a chain, which in our experiment is 'R3', and from there to access the BBS. Optionally, like in the example given bellow, we can check the node list at 'R3', before attempting to establish a link to the BBS:

```
R1:YT7MPB-1} Nodes:
BBS:YT7MPB       R3:YT7MPB-3
c r3
R1:YT7MPB-1} Connected to R3:YT7MPB-3
n
R3:YT7MPB-3} Nodes:
BBS:YT7MPB       R1:YT7MPB-1
c bbs
R3:YT7MPB-3} Connected to *BBS
[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}
                                                      (e)
PASSWORD> *****
Hello Misko!
Ch. 1     (BPQ) : YT7MPB-1  - Sat 18/04/09 19:06

1:YT7MPB BBS>
```

As shown in procedure (e) above, the first command ***c r3*** has initiated an outgoing request for transferring us from 'R1' to 'R3' and it resulted in a successful connection to the latter. Then, the second command ***n*** (which is an optional command) returned the node list of 'R3'. Finally, the third command ***c bbs*** linked us to the BBS (which then asked for password and after sending proper credentials for the particular user, brought us to the e-mail server's prompt).

As described in the first experiment, we sent two consecutive connection requests (***c r3***, ***c bbs***). Frankly speaking, that was not a complex task because of the small number of elements in the node chain. In opposite to that, if there were several repeaters in between our node and the BBS, we would have to compose many similar commands one after the other. A repetition of sending so many commands would become annoying for an average user very soon.

2.4.2 Procedure 2

With thankfulness to inventors of the amateur packet-radio networking protocols, there is a point where our repeaters' internal 'intelligence' comes to function and helps to their users. See the next experiment:

```
R1:YT7MPB-1} Nodes:
BBS:YT7MPB       R3:YT7MPB-3
c bbs
R1:YT7MPB-1} Connected to BBS:YT7MPB
[FBB-7.00i-AB1FHMRX$]
{PROTUS-4.0}
                                                      (f)
PASSWORD> *****
Hello Misko!
Ch. 1     (BPQ) : YT7MPB-1  - Sat 18/04/09 19:07

1:YT7MPB BBS>
```

At the very beginning of procedure (f), we also noticed that *BBS:YT7MPB* had been added to the node list of 'R1'. Then we issued just a single command ***c bbs***, which connected us to the BBS at 'R3'! That means, we did not have to investigate physical path(s) from 'R1' to that particular BBS, because it was enough to notice that the BBS became a part of the list of available nodes at 'R1'. The nodes did the job for us fully automatically.

If we look at updated contents (c) and (d) again, we can see that each node informs its customers about the availability of its 'first neighbor' node. In parallel with that, both 'R1' and 'R3' announced the availability of a BBS in completely the same syntax - as if the BBS is an integrated part of the both nodes. Practically speaking, it means that any visitor of either 'R1' or 'R3' can issue a single command, ***c bbs*** (or ***c yt7mpb***), which will result in connecting to the BBS. That automates the process of establishing remote connections via several 'hops', because the users of the network do not need to send more than one connecting command to the local packet-radio node.

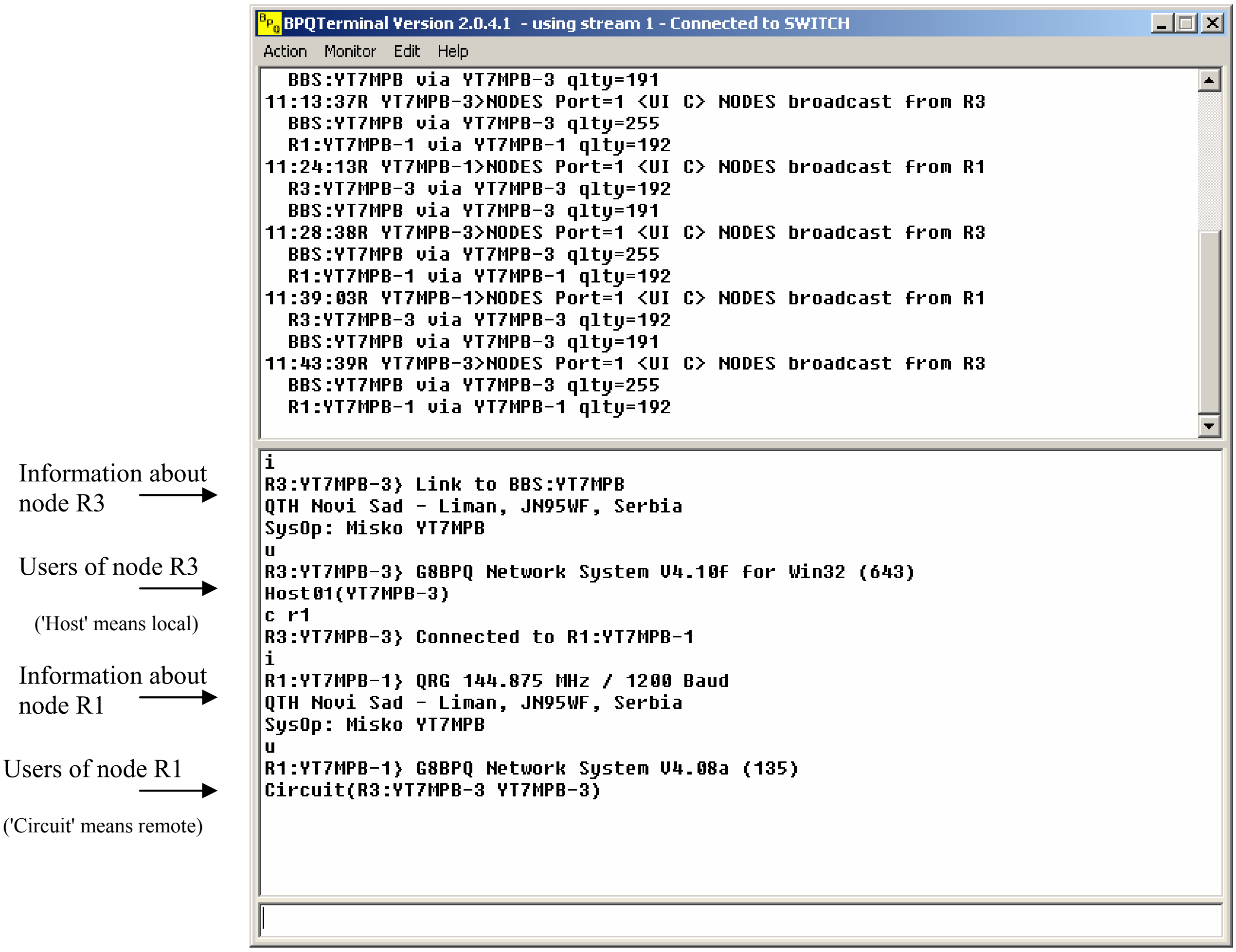
For the sake of simplicity in our examples, we investigated the automated interaction between just two nodes, where one of them had an e-mail server on the same hardware. In case of the user 'B' (bottom right side of Figure 3), he or she would initiate connection to his or her nearest repeater – in that case, it would be the node 'R2'. According to Fig. 3, the user 'B' would have a slightly more complex job to do than the user 'A', because there are three nodes in between 'B' and the BBS. Following the first scenario, the 'nodes' list of 'R2' would also become populated with the alias/callsign pairs of the neighboring nodes – probably including *BBS:YT7MPB*. If the latter happens, then the user 'B' could also command just ***c bbs***.

Following the same principle, and in cases of having a large number of active nodes in a regional 'packet-radio' network, the 'nodes' and 'routes' lists can quickly overpopulate with too many entries. In addition, the radio amateurs tend to explore all available resources they can find and that easily choke the network traffic. To prevent distant nodes from appearing in the local 'nodes' lists, some rules apply. For example, there are parameters within the node-networking software that regulate the local 'visibility' of distant nodes in a long repeater 'chain'. That parameter is not only useful for the plain nodes. It is even more important for the bulletin board systems because their customers might easily clog the regional packet-radio network by asking, for example, a bulletin list from a distant BBS. (The idea is to motivate packet users to pull relatively long bulletin lists from their nearest e-mail servers – and not from the distant ones.)

3 Other possibilities of nodes

Although there is a significant gap between the times of creation BPQ 4.08a (1995) and BPQ32 4.10f (2008), both of them have the same set of basic commands. Besides ***n***(odes), ***r***(outes), and ***c***(onnect) - explained earlier, the nodes provide opportunities for checking who is online at the moment, or who was there before and when, or what working channels are available at the particular node etcetera. The bottom half of Figure 7 – the input/output window, gives you some ideas about what a user can do when approach an amateur radio repeater. (The upper part – the monitoring window, shows the broadcast traffic between two neighboring nodes. As explained earlier, the received broadcast data populates the node-lists at both sides of a link.)

The best way to start exploring a new node is to send ***?*** or ***i***(nfo). The command ***?*** is functional with almost all repeater programs nowadays and it returns a list of available commands. If you connect to a previously unknown relay, you might be interested in learning about other users of that particular node. For example, to check "Who is there" at the moment, you will send the command ***u***(sers). To see "Who was there" earlier the same day or the previous day(s), you can send ***m***(y)***h***(eard) <#port>. In order to learn about the available ports (i.e. working channels of that node), you have the command ***p***(orts), and so on.


BPQTerminal Version 2.0.4.1 - using stream 1 - Connected to SWITCH
Action
Monitor
Edit
Help
Information about node R3
Users of node R3
('Host' means local)
Information about node R1
Users of node R1
('Circuit' means remote)

```
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:13:37R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
11:24:13R YT7MPB-1>NODES Port=1 <UI C> NODES broadcast from R1
  R3:YT7MPB-3 via YT7MPB-3 qlty=192
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:28:38R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
11:39:03R YT7MPB-1>NODES Port=1 <UI C> NODES broadcast from R1
  R3:YT7MPB-3 via YT7MPB-3 qlty=192
  BBS:YT7MPB via YT7MPB-3 qlty=191
11:43:39R YT7MPB-3>NODES Port=1 <UI C> NODES broadcast from R3
  BBS:YT7MPB via YT7MPB-3 qlty=255
  R1:YT7MPB-1 via YT7MPB-1 qlty=192
```

```
i
R3:YT7MPB-3} Link to BBS:YT7MPB
QTH Novi Sad - Liman, JN95WF, Serbia
SysOp: Misko YT7MPB
u
R3:YT7MPB-3} G8BPQ Network System V4.10f for Win32 (643)
Host01(YT7MPB-3)
c r1
R3:YT7MPB-3} Connected to R1:YT7MPB-1
i
R1:YT7MPB-1} QRG 144.875 MHz / 1200 Baud
QTH Novi Sad - Liman, JN95WF, Serbia
SysOp: Misko YT7MPB
u
R1:YT7MPB-1} G8BPQ Network System V4.08a (135)
Circuit(R3:YT7MPB-3 YT7MPB-3)
```

Fig.7. Node 'R3' administrative console: Monitoring (upper part); Communication (lower part)

BPQ 4.10f for Windows, which we used for 'R3' node, supports incoming connection requests by telnet protocol. That gives another advantage of experimenting in schools' computer rooms, because students' workstations do not require installing any additional end-user software. It means that any OS-included tool, capable to handle TCPIP traffic will fit – if the system administrator activates *BPQTelnetServer*. In that case, every predefined user would be capable to 'connect' to the node from the LAN and authenticate by a predefined user name and password.

3 Discussion and conclusion

The examples given in this paper intend to motivate newcomers to experiment with publicly available computer programs and to consider eventual activating and maintaining of a new network node. In this work, we used pretty old and, why not to say, obsolete computers. Nevertheless, we have performed the experiments smoothly. On the other side, our successful simulations proved that it is always advisable to explore new horizons by exploiting an existing laboratory setup. That means any prospective communicator who may want to reproduce those tests in his or her school is by no means obliged to purchase any new equipment! In fact, if an educational institution already has a couple of spare computers and wires them in a network, there are many opportunities for innovations like 'amateur radio in school'.

If there is a dedicated computer lab available, even better, because you can install one or two servers to run BPQ program (or the other software of that kind) and offer your students 'connecting' each other's workstations via new 'nodes'. They will become interested in exploring who of their peers is visiting, or was visiting this or that digital repeater. By occasional, temporary connections of the 'node' computer to the Internet, you will become a part of the international packet-radio network before you even install any radio-transmitting device!

Another advantage of experimenting with software-based nodes in a (closed) LAN is an ability to explore capacities and restrictions of a computer program – before you put it into a 'real' function. In the same time, you would avoid potential criticism that might come from the broader amateur radio population, because of usual beginner's errors in a system configuration.

Regardless you are a teacher or student you can perform the amateur packet radio experiments on a daily basis. After you accumulate enough enthusiasm about the amateur radio communications, you can easily add the first radio-transmitting device. Let it be a non-expensive VHF 'handy-talkie' radio (abbr. HT), like at our 'R1' node. Be careful with your radio's ability to send signals strong enough that the other stations can hear, so to avoid your HT becoming redefined as "HT" (= hidden terminal), [4]. Also related to financing new nodes, there is no need to purchase any expensive and complex data controllers of the TNC type – if you do not plan to run a 'hardware' repeater soon.

Practical exhibitions of computer-based digital nodes and bulletin board systems can be performed during specialized summer schools on the amateur radio computing, which are proposed as an added value to existing schools of similar kind – primarily those that cover various aspects of implementing new technology and innovations in computer science education, or as a separate events, [5]. If you decide to organize such an activity within your school premises, it will give you a chance for additional promotion of your institution on the air! That will motivate potential candidates for students to get a feel about the readiness of your institution to implement alternative teaching methods and pay attention to inventions within your computer labs.

Having in mind a fact that summer schools usually occur during a vacation period of the year and sometimes offer recreational activities like hitchhiking or walking through the forests or camping, you can supply more hand-held radios and laptop computers and give your students a chance to communicate with the node(s) in a schoolyard from a remote location. In addition, you can give the members of your 'expedition'-team an ad-hoc task of setting up a temporary radio-relay device at a remote location, in a rural area, or similar. In that case, you as an educator would have an immediate opportunity to check if the students completed their task properly and instruct them (by radio) about eventual relocating the unit, or reconfiguring its working parameters, etcetera. The experience in sending and receiving packet-radio messages from portable stations might help one day in saving human lives and properties after natural disasters like floods, fires, tornadoes and so on. In that direction, simulated emergency tests ensure acquiring new communicating skills, [6].

Although independent or integrated summer and winter schools are the best way to learn basic radio procedures in computer-based communications – by doing so in a slow and relaxing pace, their organizing obviously requires more efforts and materials. As an alternative, we offer another approach: To insert the school into a conference! The best events for that purpose are those so-called multi-conferences that usually take 5-7 days, so the school can develop in daily sessions of 1-2 hours per day. In order to avoid administrative procedures related to transporting radio-transmitting devices via national borders, the organizers should contact local amateur radio groups (clubs, unions) around the conference venue and offer them taking part in practical work – by using their own equipment.

By implementing amateur radio communications within our educational environment, we give our students another opportunity for on-line studying. We can also interconnect our nodes and bulletin board systems by the Internet so all of them combined would serve as a starting knowledge base. The 'ham-radio-equipped' students (and teachers too) will continue to pull available information from those mixed resources and communicate with their peers – even without fixed or mobile telephony or Internet service providers! Although they do not belong yet to the pool of 'classical' ways of e-learning methods, the amateur radio communications became their valuable supplement. By using them in parallel with more common approaches, we all can achieve better results in teaching and learning. Recent reports claim that students in on-line learning conditions perform better than those receiving traditional face-to-face instructions, [7]. Lohr (2009) finds that "the real promise of online education … is providing learning experiences that are more tailored to individual students than is possible in classrooms. That enables more learning by doing, which many students find more engaging and useful." In a conclusion, we can add that 'learning by doing' amateur radio is bringing even more joy because we use our own, locally constructed and maintained communication infrastructure. In this paper, we focused on equipping more scholars and students with exciting amateur radio skills, as a way to increase their motivation to teach engineering and improve technological literacy through that popular hobby. We believe that the amateur radio communications deserve to become a significant portion within the classroom practice.

However, besides the educational institutions, we strongly encourage other participants in the world of communications to take part in those activities. The existing amateur digital systems need continual improvements along with broader popularization. Remember that node simulations performed in home or at work (if suitable) do not require any investment in radio equipment and accessories (antennas, cables, power supply, etc).

Final tips and reminders:

- If your institution (or home) is located in a high building, we strongly recommend you to start experimenting with digital amateur radio repeaters – 'nodes'. The higher geographic position you have – the larger distance in communication you will get! Your younger practitioners will be satisfied with services your node can offer – particularly if you live in a developing country that suffers from a shortage of technological resources, [8].

- You should start with a simple node, like our 'R1'. In the next iteration, you can install something like our 'R3' networking relay – either with or without a bulletin board system (BBS). Then, you can choose if you want to leave both nodes working together, or to leave the simpler one ('R1') to run permanently and have the more complex one ('R3' + BBS) working occasionally, etc.

- Keep an eye on daily routines in activities of your new users, to get familiar with their behavior. Keep them informed about the best practices you have examined within the radio-relay facilities that are under your control. Be aware that there are many safety measures to secure amateur radio repeaters and bulletin boards against unauthorized use – especially the high privileged accounts. System administrators are responsible for their systems and should perform suitable protective activities to avoid any potential misuse of their systems, [9].

- Those 'sysops' (= system operators) who have reliable connectivity to the Internet in their school labs, engineering or other companies, as well as in their households, are invited to add and configure TCPIP ports to their experimental nodes. By having an Internet 'tunnel' between two distant nodes, you can get a feel that you have managed to acquire an intercontinental link with a remote station, but without expensive high frequency (HF) radios that operate at international short waves, [10].

- If you, as a teacher or student, consider implementing the amateur radio technology in your educational environment, you do not need to have computer-expert knowledge as a prerequisite! In fact, if you are ready to read operating manuals and follow instructions, you can easily become an active *packeteer*. "Like any other facet of Amateur Radio, you're going to have to learn some new concepts, but that's part of the enjoyment", says Steve, WB8IMY, [6].

References:

[1] M. Skoric, The perspectives of the Amateur University Networks – AMUNETs, *WSEAS Transactions on Communications*, Issue 9, Volume 4, September 2005, pp 834-845.

[2] M. Skoric, The new amateur radio university network – AMUNET (Part 2), 10th WSEAS Int. Conf. on Computers, *Proceedings*, Athens, 2006.

[3] M. Skoric, *FBB Packet-radio BBS mini-HOWTO*, The Linux Documentation Project, http://tldp.org/HOWTO/FBB.html, 2000-2009.

[4] G. Jones, *Packet Radio: What? Why? How?*, TAPR, 1996.

[5] M. Skoric, Summer Schools on the Amateur Radio Computing, 12th ACM ITiCSE 2007, *Proceedings*, 2007.

[6] S. Ford. *Your Packet Companion*. ARRL, 1995.

[7] S. Lohr, Study Finds That Online Education Beats the Classroom, *New York Times*, August 19, 2009, http://bits.blogs.nytimes.com/2009/08/19/study-finds-that-online-education-beats-the-classroom/

[8] M. Skoric, The Amateur Radio as a Learning Technology in Developing Countries, *Proceedings*, IEEE ICALT-TEDC, Joensuu, Finland, 2004.

[9] M. Skoric, The perspectives of the Amateur University Networks – AMUNETs (Part 2), *WSEAS Transactions on Communications*, Issue 6, Volume 7, June 2008, pp 616-626.

[10] S. Ford. *Your HF Digital Companion*. ARRL, 1995.