Microchip MiWi and P2P IEEE 802.15.4 Reference Files

Dr. Richard Wall – Professor
Department of Electrical and Computer Engineering
University of Idaho
Moscow, ID 83844-1023
December 19, 2012
rwall@uidaho.edu

---

1. **ZigBee and Wireless Standards - see**

   http://www.stg.com/wireless/ZigBee_comp.html.

   https://sites.google.com/site/xbeetutorial/

2. Basic Concepts

a. Definition of: **ZigBee**

"A wireless network used for home, building and industrial control. It conforms to the IEEE 802.15.4 wireless standard for low data rate networks. With a maximum speed of 250 Kbps at 2.4 GHz, ZigBee is slower than Wi-Fi and Bluetooth, but is designed for low power so that batteries can last for months and years. The typical ZigBee transmission range is roughly 50 meters, but that can vary greatly depending on temperature, humidity and air quality.

b. Zigzag Like a Bee

Although ZigBee networks can be configured in star, peer-to-peer and mesh topologies, it is the mesh network from which ZigBee was named. A ZigBee mesh provides multiple pathways from device to device (like the Internet) and eliminates a single point of failure. If nodes go down or are removed, ZigBee devices can "zig" and "zag" through the network to their destination like a bumblebee.

c. **Lots of Bees**[1]

ZigBee networks are simple control networks that periodically send small packets from sensors to regulate lights, motors and other equipment. A large building can have tens of thousands of ZigBee nodes; a home could have a hundred or more. In fact, ZigBee can address more than a thousand quadrillion devices (surely enough for the gadget fanatic's apartment!).

d. **Reduced Function and Full Function**

ZigBee uses two types of devices. Reduced-function devices (RFDs) are sensors that communicate with full-function devices (FFDs). FFDs are complex nodes that conform to the full 802.15.4 standard and can serve as routers. All devices can be implemented with low-cost, 8-bit microcontrollers (MCUs) that derive their power from two AAA batteries. For more information, visit the ZigBee Alliance at www.zigbee.org ."

3. **MiWi vs Zigbee**[2]

MiWi and **MiWi P2P**[3] are proprietary wireless protocols designed by Microchip Technology that uses small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). It is designed for low data transmission rates and short distance, cost constrained networks, such as industrial monitoring and control, home and building automation, remote control, low-power wireless sensors, lighting control and automated meter reading.

The MiWi protocols are supported on certain Microchip PIC and dsPIC microcontrollers. When developing for these platforms, proprietary SDKs and hardware development tools, such as the ZENA wireless packet sniffer, may be used.

The Microchip **ZENA** (or formerly, **Z**igbee **E**nhanced **N**etwork **A**nalyzer) is a wireless packet sniffer and network analyzer following the IEEE 802.15.4 specification on the 2.4 GHz band. The ZENA analyzer supports both the ZigBee and MiWi protocols. Accompanying software can analyze network traffic and graphically display decoded packets. It can also display the network topology and the messages as they flow through the network. With the provided key of the network, data on encrypted MiWi networks can be sniffed and viewed as well

a. **Zigbee overview**
b. **ZIGBEE WIRELESS NETWORKING,** Drew Gislason, Newnew Publishers, ISBN 978-0-7506-8597-9

[1] http://www.pcmag.com/encyclopedia_term/0,1237,t=ZigBee&i=55212,00.asp

[2] en.wikipedia.org/wiki/MiWi

[3] P2P Protocol - Reference Microchip AN1024

c. **MiWi and P2P Applications Notes**

AN965 Microchip Stack for the Zigbee Protocol

AN1066 Microchip MiWi Wireless Network Protocol Stack

AN1204 Microchip MiWi P2P Wireless Protocol

AN1283 Microchip Wireless (MiWi) Media Access Controller - MiMAC

AN1284 Microchip Wireless (MiWi) Applications Programming Interface MiApp

MiWi P2P Stack Application Programming Interfaces

MiWi PRO Application Notes

d. **Hardware Support**

MRF24J40MA 2.4 GHz RF Transceiver

MRF24J40 Data Sheet

Digilent PModRF2

Cerebot MX7ck

Microchip ZENA Wireless Adapter User's Guide

e. **Software Support**

Microchip Libraries for Applications v2012-10-15

Microchip Application Libraries Help Files

f. **Microchip MiWi P2P Tutorial**
g. **Microchip MiWi Tutorial**
h. **ZENA installation notes**

4. Reference Designs

a. RD21a through RD21c are the projects for the three node example of the MiWi P2P protocol. The P2P (peer to peer) version depends on all nodes being in the RF range of each other. This example can be expanded to include additional nodes by changing the device identification. This involves modifying the EUI (lines 100 – 107 of ConfigApp.h) for each node. Generally, you will only need to change EUI_0 to be unique for each personal area network (PAN). Displayed identifiers in the DemoOutput.c file will also need to be changed so the correct information will reported on the display devices (LCD and Console). This is accomplished by changing the text in the string constants on lines 81, 88, 94, and 97 in the DemoOutput.c file. The identifier sent out over the IEEE 802.15.4 network is assigned the value in main.c by the declaration on line 81 for the constant BYTE AdditionalNodeID[ADDITIONAL_NODE_ID_SIZE]. In the three node example, the initialized value can be set to LIGHT (1), SWITCH_1 (2), or SWITCH_2 (3). (See lines 66-68.) See application notes listed above for additional information.

b. RD22a through RD22c are the projects for the three node example of the MiWi protocol. All nodes do not need to be in RF range of each other as in the P2P implementation.

5. Use of the ZENA MiWi and P2P Packet Sniffer

a. Launch the WDS application that has the Microchip ZENA Wireless Adapter 24 GHz MRF24J40 attached. The PC window will appear as shown in Figure 1.

b. If the Radio: box does not initial show MRF24j40, click on the drop down list and select that option.

c. Click on the drop down list for the “Tool” box and select the only other available option except for <No Device Selected>

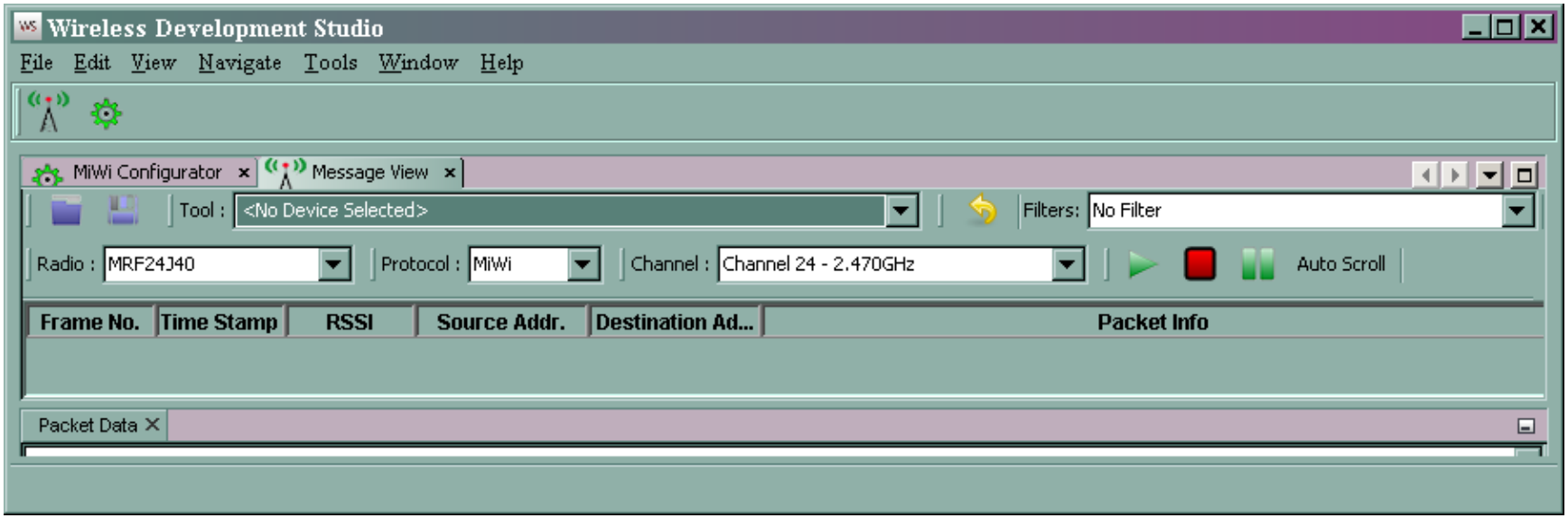

Figure 1. Initial WDS applications

d. Select the Channel to be the same as shown on the LCD or the PC terminal window after the PIC32 has bee reset. If using the LCD – look quickly because the LCD display will change after two seconds. The terminal display for Node 1 is shown in Figure 2 and for Node 2 in Figure 3.

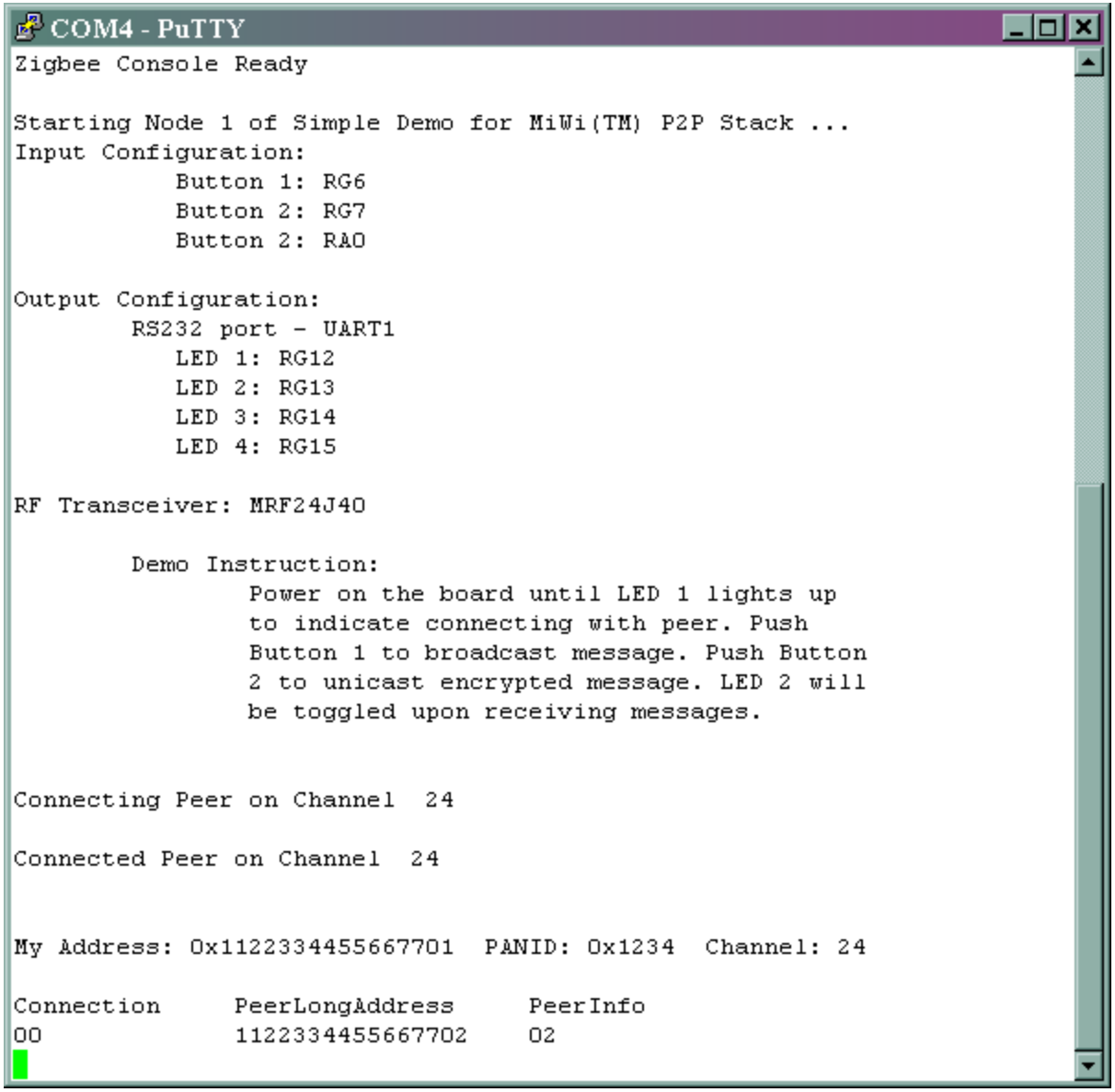

Figure 2. Terminal window after reset and connection for Node 1

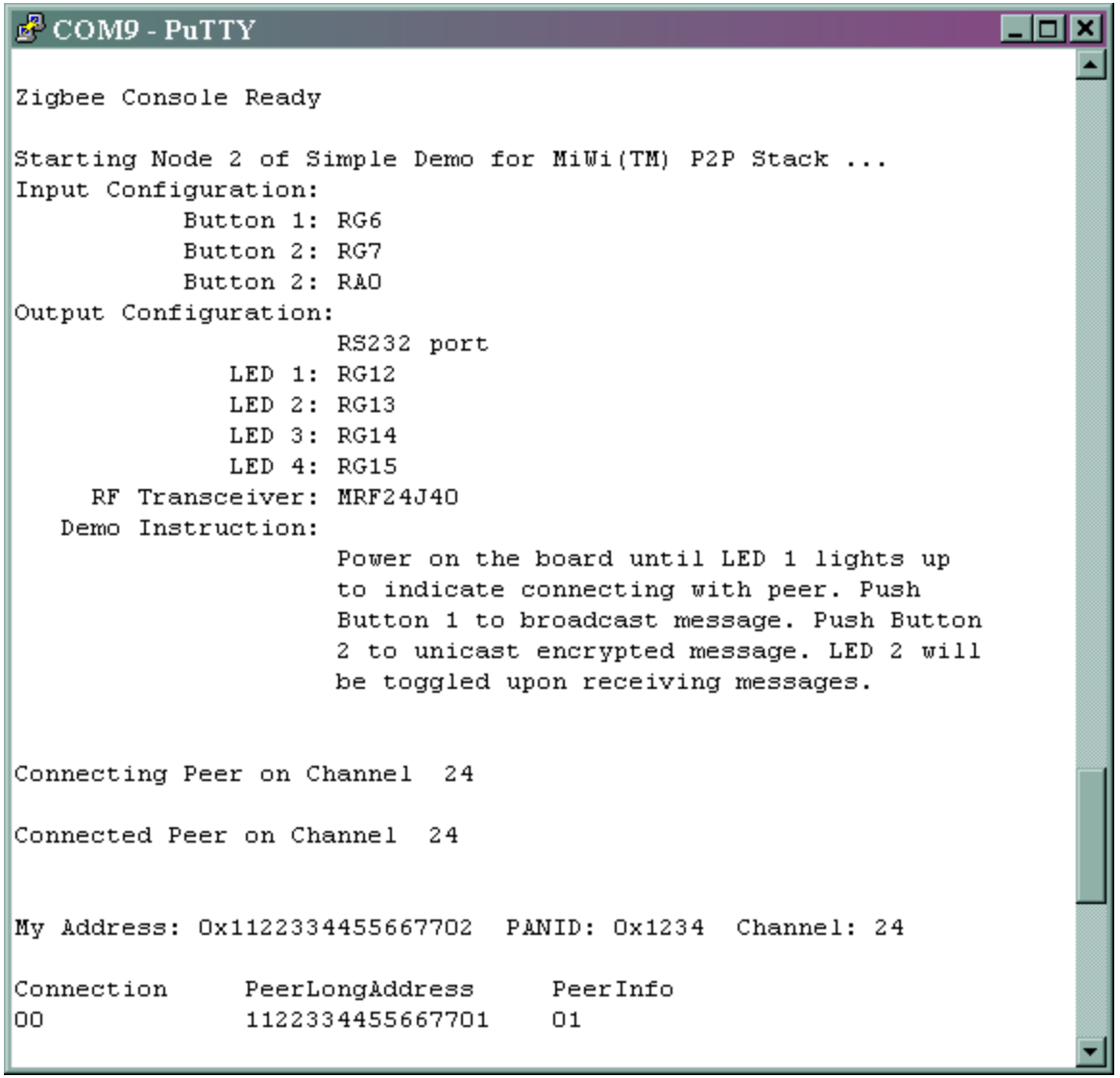

Figure 3. Terminal window after reset and connection for Node 2

e. Select either MiWi P2P or MiWi depending upon the protocol being used. Selecting the a protocol does not appear to alter the ability

f. Figure 4 Is a screen capture WDS running during the power up process of Node 2 and subsequent MiWi packets.

g. Lines 1 through 3 is the beacon request from Node 2 that searches for a node to communicate with. The decoded Line 1 Connection Request is shown in Figure 5.

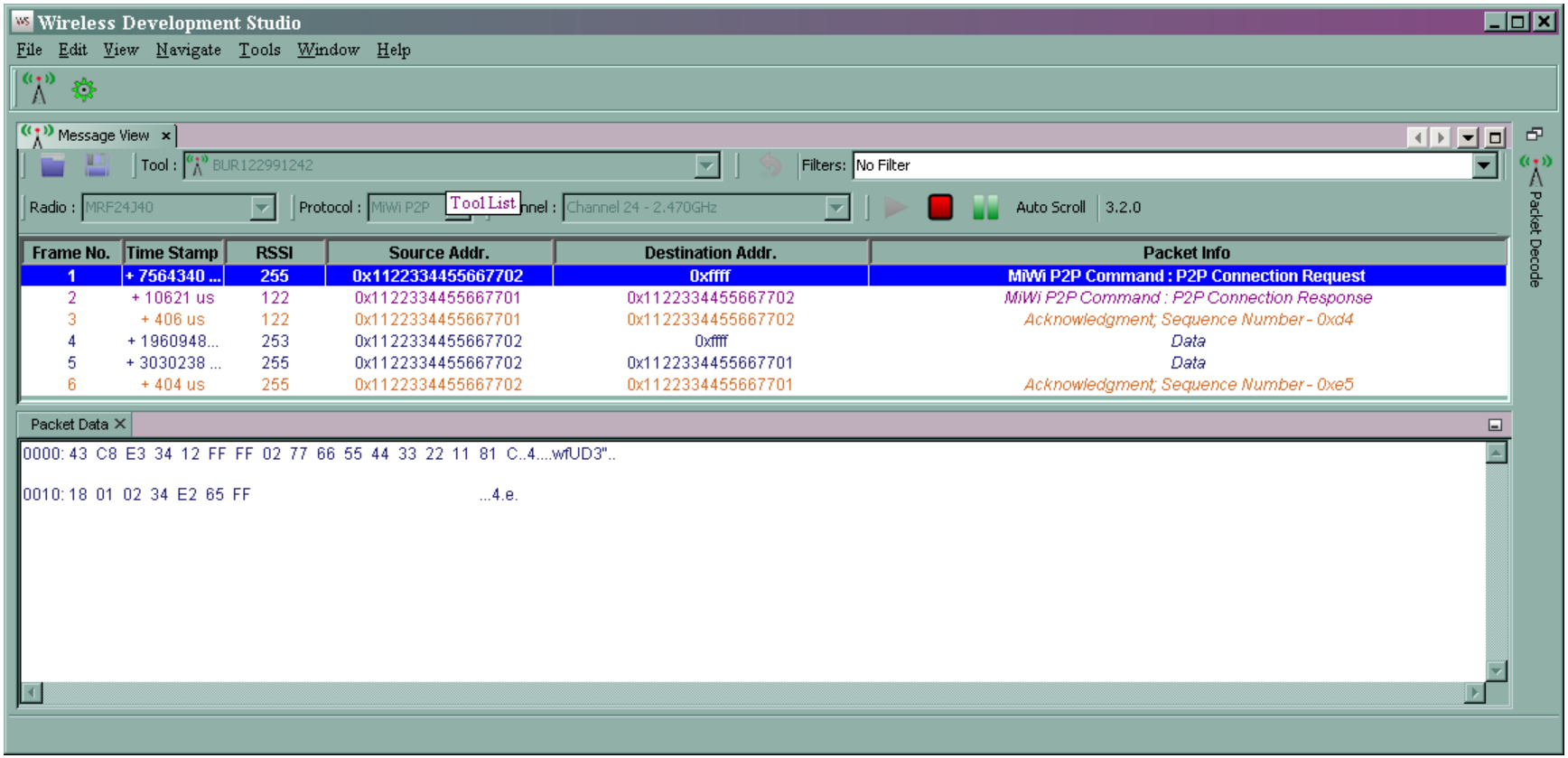

Figure 4. Screen capture of the boot sequence for Node 2 MiWi P2P connection

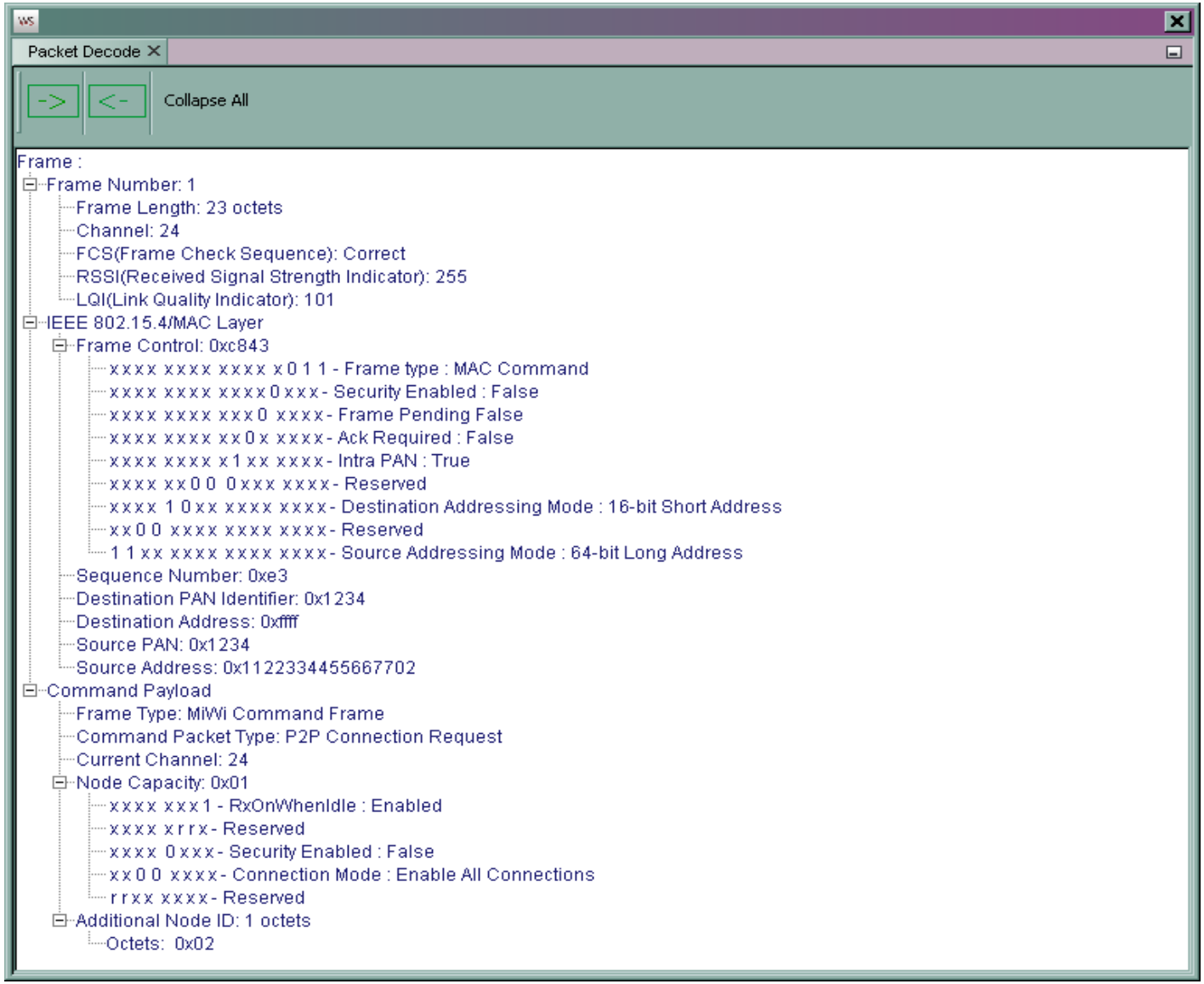

Figure 5. A decoded Connection Command (Line 1) for Node 2 reset in P2P Mode

h. Line 2 of Figure 4 shows the connection response from Node 1. The decoded packet is shown in Figure 6.

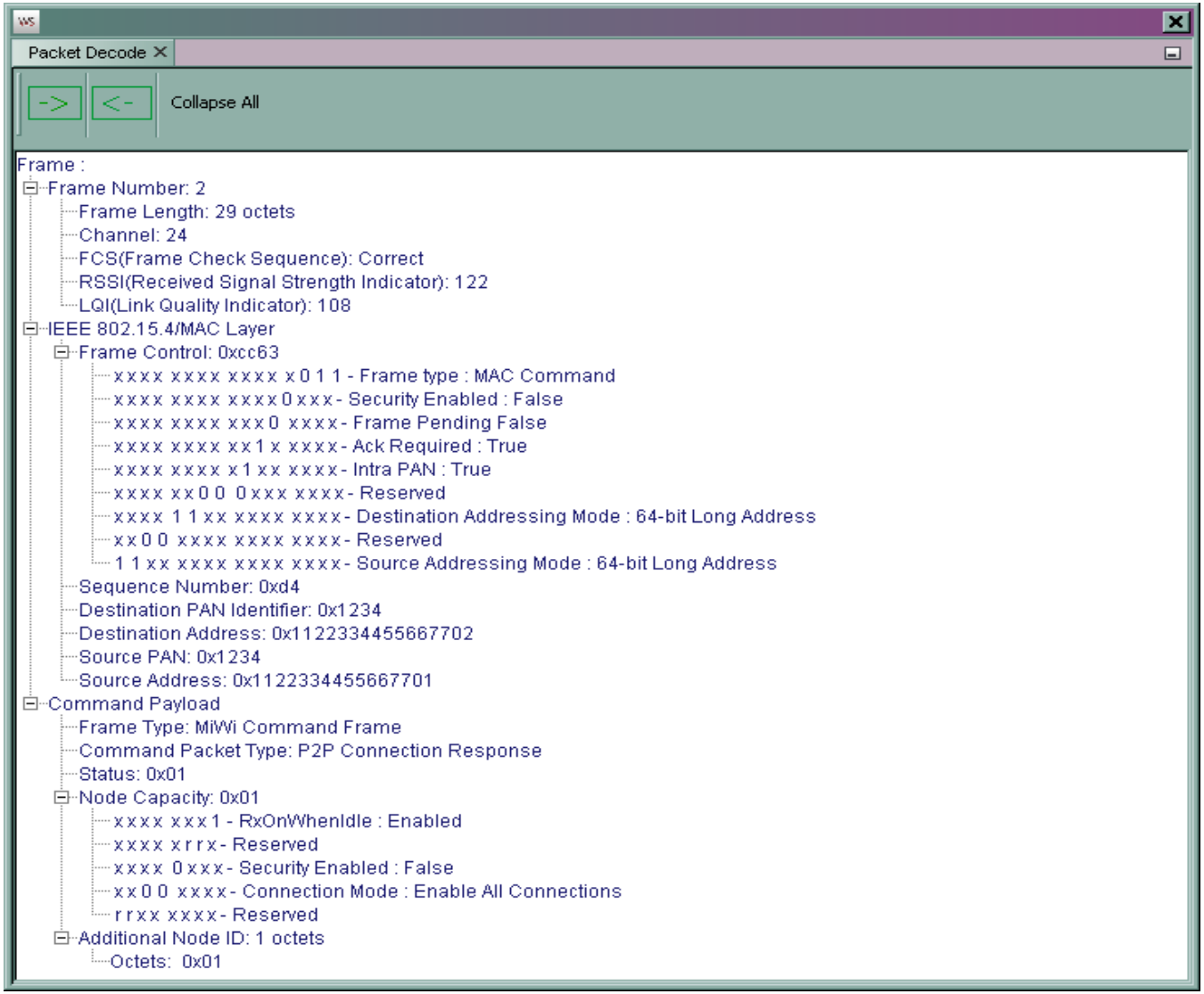

Figure 6. A decoded Connection Response (Line 2) for Node 2 reset in P2P Mode

i. Line 3 of Figure 4 shows the Acknowledgment from Node2. The decoded packet is shown in Figure 7.

Figure 7. The Connection Acknowledgement from Node 1 to Node 2

j. Lines 4 and 5 of Figure 4 shows that a Broadcast and a Unicast packet was sent from Node 2. Figure 8 shows the terminal window for Node 1in response to the two packets.

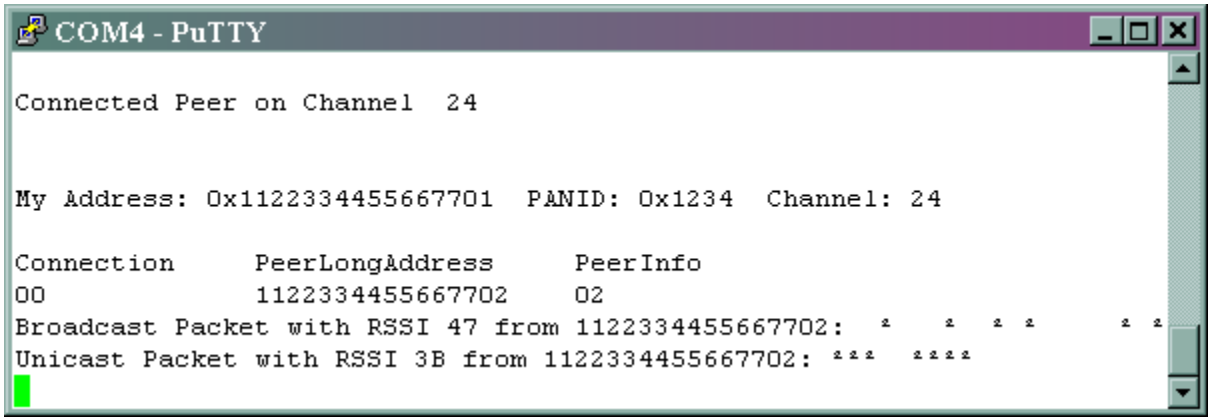

Figure 8. The terminal window for Node 1 in response to a broadcast packet followed by a unicast from Node 2.

k. Line 4 of Figure 4 shows that a Broadcast packet (all nodes on the operating channel and PANID) was sent from Node 2. The decoded packet is shown in Figure 9.

l. Line 5 of Figure 4 shows that a Unicast (Sent to a specific Node address) packet was sent from Node 2 to Node 1. The decoded packet is shown in Figure 10.

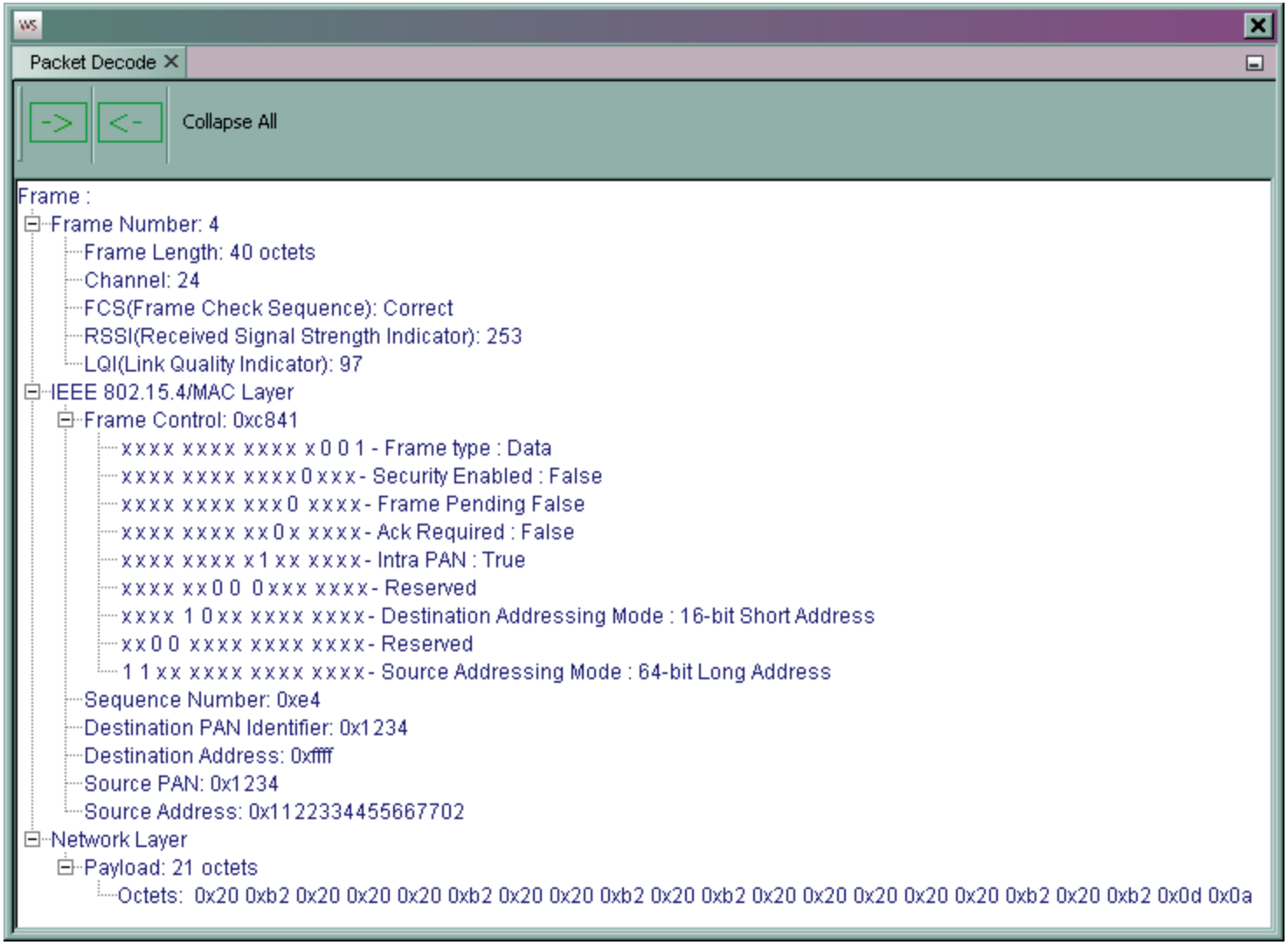

Figure 9. Decoded Broadcast packet from Node 2

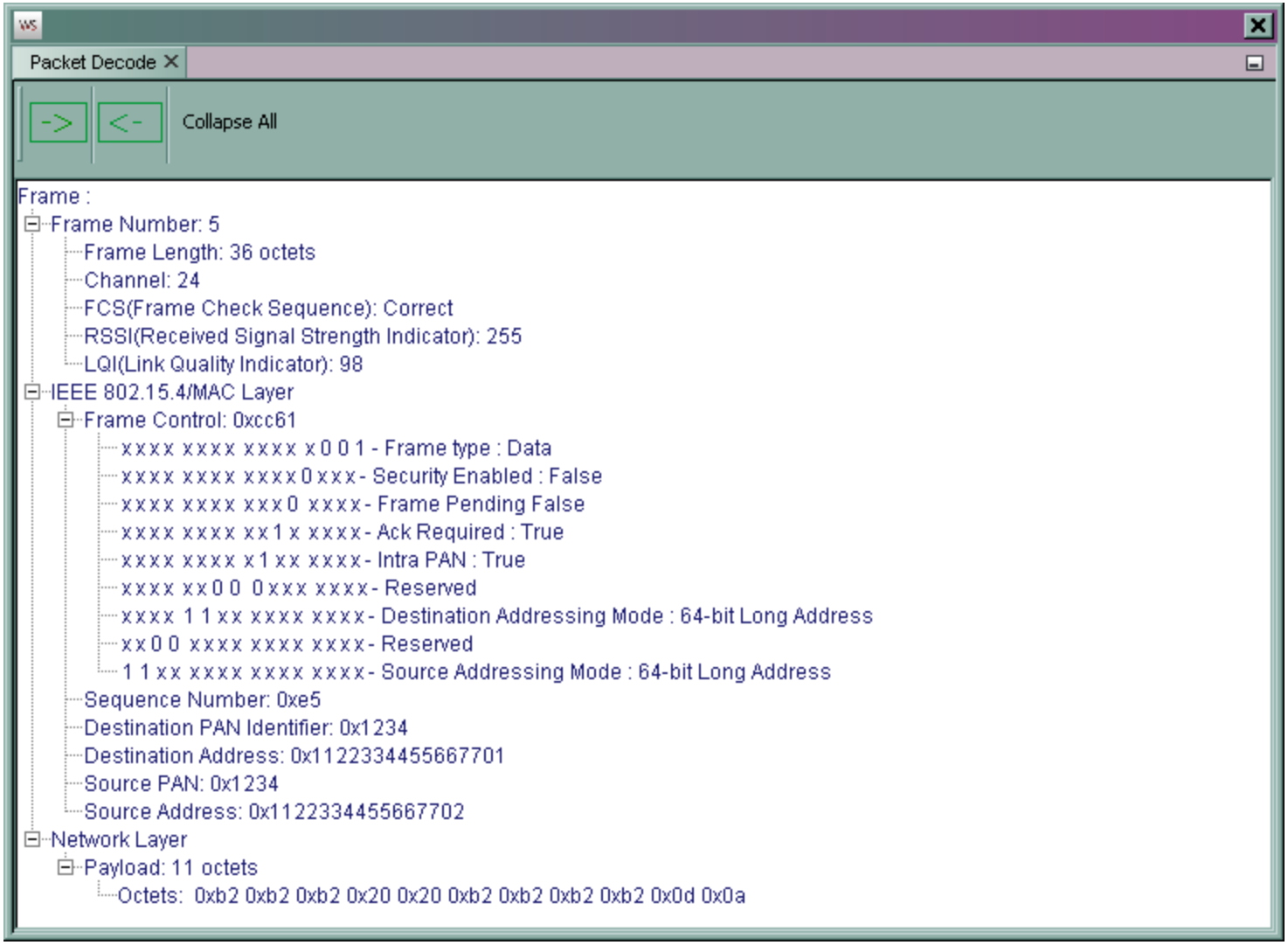

Figure 10. Decoded Unicast packet from Node 1 to Node 2

2. What is the message?
    a. For Broadcast packets – the message is revealed after 6 successive presses of Button 1.

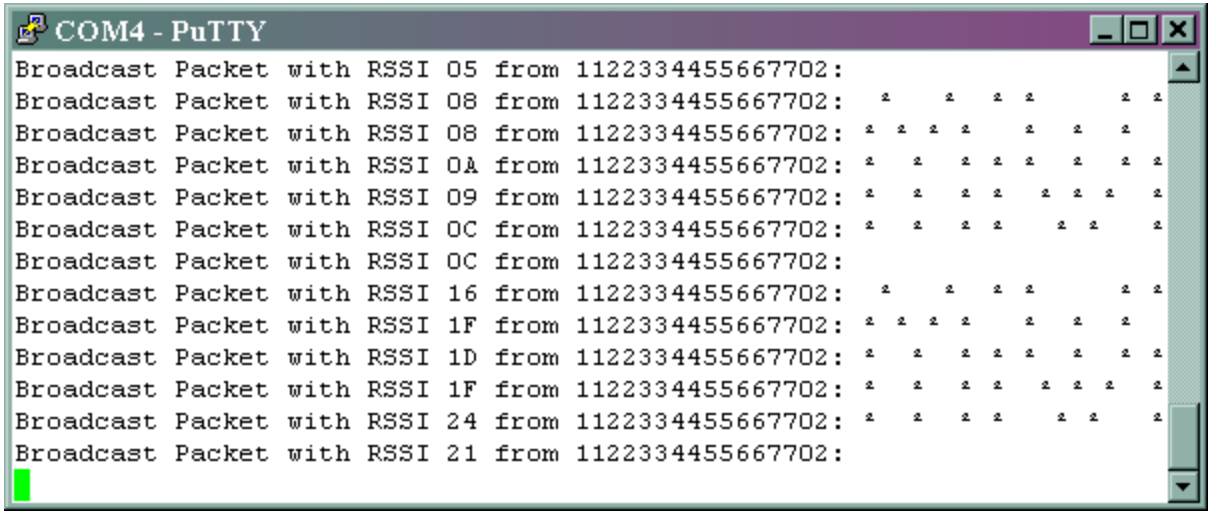

Figure 11. Brastcast messages sent from Node 2 as seen on the terminal window for Node 1. "MiWi"

b. For Unicast packets – the message is revealed after 6 successive presses of Button 2.

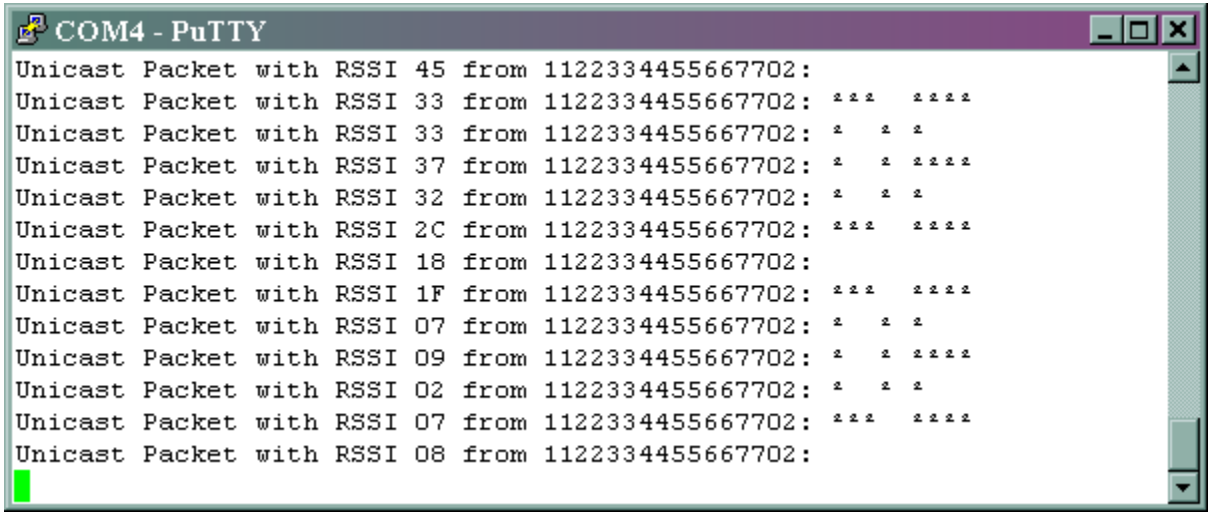

Figure 12. Unicast messages sent from Node 2 to Node 1

3. Simple Examples MiWi Configuration Options – ConfigApp.h - options are disabled by placing comment marks "//" in front of the #define statements.
    a. ENABLE_CONSOLE will enable the print out on the hyper terminal this definition is very helpful in the debugging process . Our examples use the consol so the #define ENABLE_CONSOLE statement is not commented out.
    b. HARDWARE_SPI enables the hardware SPI implementation on MCU silicon. If HARDWARE_SPI is not defined, digital I/O pins will be used to bit-bang the RF transceiver. Our application uses bit bit-bang SPI therefore HARDWARE_SPI is commented out.
    c. PROTOCOL_P2P enables the application to use MiWi P2P stack. This definition cannot be defined with PROTOCOL_MIWI. define PROTOCOL_P2P is enable for the Simple Examples
    d. PROTOCOL_MIWI enables the application to use MiWi mesh networking stack. This definition cannot be defined with PROTOCOL_P2P. #define PROTOCOL_MIWI is used for the full feature examples
    e. NWK_ROLE_COORDINATOR is not valid if PROTOCOL_P2P is defined. It specified that the node has the capability to be coordinator or PAN coordinator. This definition cannot be defined with NWK_ROLE_END_DEVICE.
    f. Definition of MRF24J40 enables the application to use Microchip MRF24J40 2.4GHz IEEE 802.15.4 compliant RF transceiver. Only one RF transceiver can be defined. The statement #define MRF24J40 is enabled.
    g. ENABLE_NETWORK_FREEZER enables the network freezer feature, which stores critical network information into non-volatile memory, so that the protocol stack can recover from power loss gracefully. The network information can be saved in data EPROM of MCU, external EEPROM or programming space, if enhanced flash is used in MCU. Network freezer feature needs definition of NVM kind to be used, which is specified in HardwareProfile.h. #define ENABLE_NETWORK_FREEZER is active for our application and uses the on board 24L256 I2C EEPROM
    h. MY_ADDRESS_LENGTH defines the size of wireless node permanent address in byte. This definition is not valid for IEEE 802.15.4
    i. compliant RF transceivers. The #define MY_ADDRESS_LENGTH 4 is active.
    j. EUI_x defines the xth byte of permanent address for the wireless node. For example, the address for Node 1 is define by:

```
#define EUI_7 0x11
#define EUI_6 0x22
#define EUI_5 0x33
#define EUI_4 0x44
#define EUI_3 0x55
#define EUI_2 0x66
#define EUI_1 0x77
#define EUI_0 0x01
```

k. TX_BUFFER_SIZE defines the maximum size of application payload which is to be transmitted - #define TX_BUFFER_SIZE 40
l. RX_BUFFER_SIZE defines the maximum size of application payload which is to be received - #define RX_BUFFER_SIZE 40
m. MY_PAN_ID defines the PAN identifier. Use 0xFFFF if prefer a random PAN ID - #define MY_PAN_ID 0x1234
n. ADDITIONAL_NODE_ID_SIZE defines the size of additional payload will be attached to the P2P Connection Request. Additional payload is the information that the devices what to share with their peers on the P2P connection. The additional payload will be defined by the application and defined in main.c - #define ADDITIONAL_NODE_ID_SIZE 1
o. P2P_CONNECTION_SIZE defines the maximum P2P connections that this device allows at the same time. #define CONNECTION_SIZE 10
p. TARGET_SMALL will remove the support of inter PAN communication and other minor features to save programming space. // #define TARGET_SMALL – function is disabled
q. ENABLE_PA_LNA enable the external power amplifier and low noise amplifier on the RF board to achieve longer radio communication range. To enable PA/LNA on RF board without power amplifier and low noise amplifier may be harmful to the transceiver. //#define ENABLE_PA_LNA– function is disabled
r. ENABLE_HAND_SHAKE enables the protocol stack to hand-shake before communicating with each other. Without a handshake process, RF transceivers can only broadcast, or hardcoded the destination address to perform unicast. #define ENABLE_HAND_SHAKE – function is enabled.
s. ENABLE_SLEEP will enable the device to go to sleep and wake up from the sleep. #define ENABLE_SLEEP – function is disabled
t. ENABLE_ED_SCAN will enable the device to do an energy detection scan to find out the channel with least noise and operate on that channel. #define ENABLE_ED_SCAN – function is enable on full featured examples
u. ENABLE_ACTIVE_SCAN will enable the device to do an active scan to detect current existing connection. #define ENABLE_ACTIVE_SCAN – function is enable on full featured examples
v. ENABLE_SECURITY will enable the device to encrypt and decrypt information transferred. #define ENABLE_SECURITY – function is currently disabled
w. ENABLE_INDIRECT_MESSAGE will enable the device to store the packets for the sleeping devices temporally until they wake up and ask for the messages. #define ENABLE_INDIRECT_MESSAGE – function is currently disabled
x. ENABLE_BROADCAST will enable the device to broadcast messages for the sleeping devices until they wake up and ask for the messages. #define ENABLE_BROADCAST – function is currently disabled
y. RFD_WAKEUP_INTERVAL defines the wake up interval for RFDs in second. This definition is for the FFD devices to calculated various timeout. RFD depends on the setting of the watchdog timer to wake up, thus this definition is not used. #define RFD_WAKEUP_INTERVAL 8 – functionality is enabled when appropriate

z. ENABLE_FREQUENCY_AGILITY will enable the device to change operating channel to bypass the sudden change of noise. #define ENABLE_FREQUENCY_AGILITY – functionality is enable for full featured examples.