Table of Contents

1 Introduction

The nRoute protocol is used in communication between a node and a PC over the serial port. The protocol allows data transfer and RF configuration. The protocol fields allow extensions, such as new configuration options and address types.

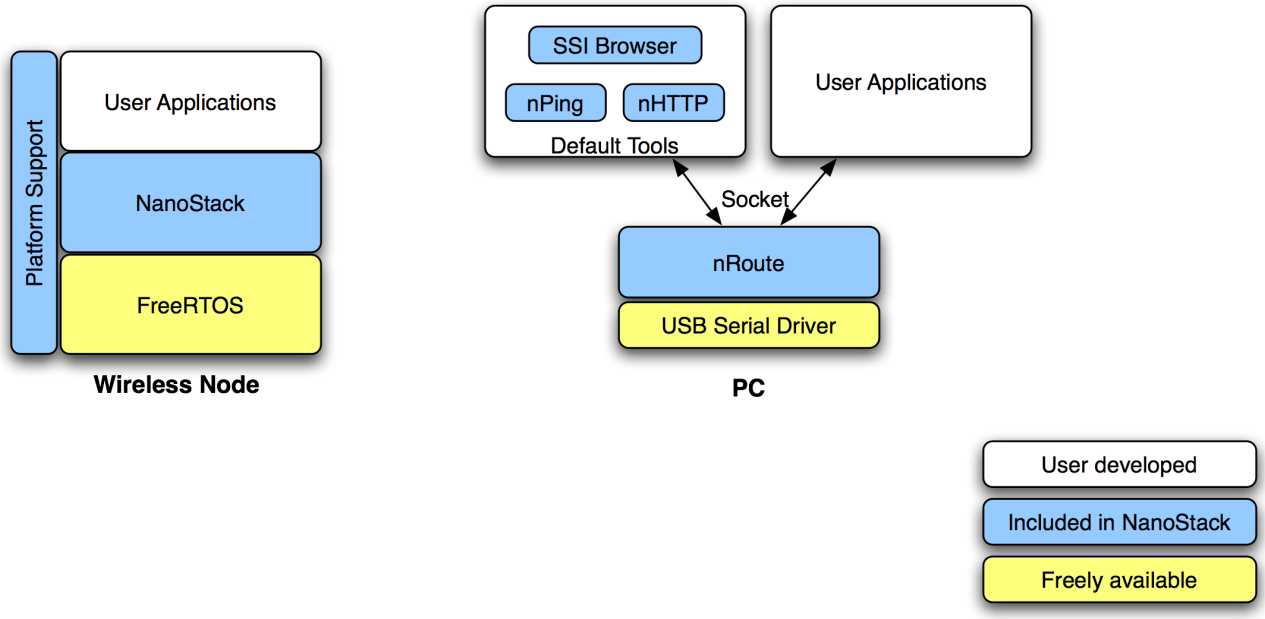

Figure 1: The NanoStack protocol architecture.

2 Packet structure

In the PC, the example implementation of nRoute Protocol (hereafter nRP) is included in nRoute. In the Micro series NanoStack implementation the code is implemented in the nrp.c driver module[1].

The nRP packets contain a header to allow identification of nRP traffic. There are two types of packets: data and configuration packets. After the header, the packets are constructed as “tag, length, value” type fields. This allows different protocol versions to parse packets and skip parsing fields that are not understood by the receiving software version. The Tag field uses only 7 lowest bits of the octet. The highest bit is used to indicate the beginning of the last Tag-Length-Value triplet by setting the bit to 1.

In addition, nRP has a debug data option allowing simultaneous debugging and nRP operation. The nRP debug packets do not follow the “tag, length, value” format and use a terminating null character to indicate end of packet.

1 The driver is found in the platform directory in the MCU software tree.

Table 1: NRP packet structure.

Field name	Header	Version/Type	Tag	Length	Value	...
Bytes	3	1 (4 bit fields)	1	2	Field size	...
Value	“NRP”	version|type	Field ID	Field size	Field data	...

nRP data fields are always in most significant bit (MSB) first “network byte order”.

2.1 Data packet fields

Data packets use type ID 0 and carry data packets from the RF to the host PC or vice versa.

Table 2: nRP data packet fields.

Field tag	Length	Value	Field tag	Length	Value
0x00	n	Packet data	**0x01**	1	Protocol
0x02	n	Source address	**0x03**	n	Destination address
0x04	2	Source port	**0x05**	2	Destination port
0x06	2	Signal level (dBm)	**0x07**	2	Packet sequence ID

2.1.1 Packet data field

The packet data field contains the actual payload data **excluding** the headers for the protocol. Data required to construct headers must be present in additional fields, such as port number and address fields.

2.1.2 Protocol field

The Protocol field defines the protocol type the Packet data contains. The type is indicated by a single octet. Definitions are listed in Table 3.

Type	Description
0x00	IEEE 802.15.4 MAC data
0x01	nUDP †
0x02	6lowpan †
0x03	ZigBee †
0x04	
0x05	
0x06	

Table 3: nRP protocol types

2.1.3 Address fields

The address field contains an address type identifier in the first byte, followed by the address data in network byte order. The length in the table shows field length, that is

† Not implemented in NanoStack v0.9

address size + 1 for type identifier.

2.1.4 Signal level field

The signal level field contains a 16-bit signed integer value representing the receiver signal level in dBm.

2.1.5 Packet sequence ID field

If a nRP data packet contains the **0x07** field and the corresponding Packet sequence ID then the sender expects the receiver to inform if the data was successfully sent to the host specified by the **0x03** and **0x05** fields (Destination address and port number).

If the receiver is not able to tell whether the data reached its destination a reply with value **0x0** for field **0x07** is sent. If the receiver knows that the data did not reach destination then the value of field **0x07** will be **0xffff**. If the data was successfully sent to destination and the receiver knows this then the value of the **0x07** will be the same which was used by the initial sender.

Table 4: nRP address types.

Type	Length	Description
0x00	7	6-byte IEEE hardware identifier (such as Ethernet card MAC) †
0x01	9	802.15.4 device long address (64 bits)
0x02	3	802.15.4 PAN ID (16 bits) †
0x03	3	802.15.4 short address (16 bits) †
0x10	n+1	IP address (name string) †
0x11	5	IP version 4 address (32 bits) †
0x12	17	IP version 6 address (128 bits) †

†

2.2 Configuration query packets

Configuration query packets use type ID 1 and carry configuration data from the RF to the host PC or vice versa.

Table 5: nRP configuration query ID's.

Field tag	Length	Query	Field tag	Length	Query
0x00	0	get nRP version	**0x01**	0	get RF network IDs
0x02	0	get protocol ID's	**0x03**	0	packet subscribe
0x04	0	get protocol stacks	**0x05**	1	get protocol address
0x06	n	Signal level per address	**0x07**	0	Module reset

2.2.1 Version query (0x00)

The version query does not require additional parameters. The module shall respond

† Not implemented in NanoStack v0.9

with Version configuration reply (0x00).

2.2.2 RF ID query (0x01)

The ID query does not require additional parameters. The module shall respond with RF ID configuration reply (0x01).

2.2.3 Protocol ID query (0x02)

The protocol query does not require additional parameters. The module shall respond with Protocol ID configuration reply (0x02).

2.2.4 Packet Subscribe query (0x03)

The subscribe query requires the following data tags:

- Protocol ID tag
- Source address tag (optional)
- Source and/or destination port tag (optional)

The module shall respond with subscribe configuration reply (0x03).

Subscribe query with only the protocol ID specified will result in receiving all packets using this protocol. Address and port tags are used as additional filters.

2.2.5 Module reset (0x07)

Reset the nRP module. Requires no parameters. Will execute hardware reset on the module.

2.3 Configuration reply packets

Configuration reply packets use type ID 2 (BIN:0010) and can be sent either from RF or PC.

Table 6: nRP configuration packet fields.

Field tag	Length	Reply	Field tag	Length	Reply
0x00	1	nRP version	**0x01**	n*1	get RF network IDs
0x02	n*1	get protocol ID's	**0x03**	1	packet subscribe
0x04	0	get protocol stacks	**0x05**	1	get protocol address
0x06	n	Signal level per address	**0x07**	0	

2.3.1 Version reply (0x00)

The version reply contains 1 byte of data, containing the nRP version supported in the following format:

Field name	Protocol version	Interface version
Bits	4	4
Value	NRP protocol version	Interface version

The version reply does not provide additional data tags.

2.3.2 RF ID reply (0x01)

The RF ID reply contains a sequence of RF type ID's(1 byte each), corresponding to the index number, i.e. the first byte is the type ID of RF interface 0, the second is ID of interface 1, etc.

The RF ID reply does not provide additional data tags.

2.3.3 Protocol ID reply (0x02)

The protocol ID reply contains a sequence of protocol ID's(1 byte each). These protocol are supported by the nRP module.

The Protocol ID reply does not provide additional data tags.

2.3.4 Packet Subscribe reply (0x03)

The subscribe query contains a single byte containing a signed value. If the value is positive, the return value represents the identifier of the requested packet filter. If the value is negative, the filter could not be enabled.

2.4 nRouted configuration packet fields

The nRP packet format is reused by a remote host connected to the nRoute daemon to configure the packets that are to be relayed to the host. This special version of the nRP packet format is recognized by the type ID: 3. The packet format is very close to that of the nRP data packet, but with only a subset of the allowed fields.

Table 7: nRP nRouted configuration packet fields.

Field tag	Length	Value	Field tag	Length	Value
0x00	1	Protocol	**0x01**	n	Source address
0x02	n	Destination address	**0x03**	2	Source port
0x04	2	Destination port	**0x05**	n	Reply string

The Field tag **0x05** is reserved to be used as a reply by the nRoute daemon to the client application and thus can never be used in same nRP packet with any of the tags **0x00** through **0x04**. After the nRoute daemon has parsed the configuration packet from the client it replies by sending a nRP packet that has only the **0x05** tag. If there was no errors during parsing the Length will be 2 bytes and the Value will be 'OK'. In case of errors the Length will be 4 bytes and the Value will be 'FAIL'.

2.5 nRP MCU debug packet

A nRP packet with type ID **0x0f** is used to send debugging information from u.4 node to PC. This is extremely useful because the UART of the u.4 device (gateway dongle) is reserved to the communication with the PC and thus can not be used just for debugging in some situations.

Table 8: nRP MCU debug packet format.

Length	Value
n	C string to be printed out as is, null terminated

3 Examples

3.1 nRP initialization example

Will be added in future versions of this document.

3.2 nRP traffic example

Example data packet:

Table 9: nRP example data packet.

Field tag	Length	Value	Field tag	Length	Value
N/A	4	4e 52 50 00	**0x05**	n	Reply string
0x00	2	aa ee	**0x83**	9	01 FFFFFFFF FFFFFFFF

The first 4 bytes are the header: NRP (in ASCII) + **0x00** for version 0 data packet. The next field defines the protocol: 802.15.4 MAC packet. Field **0x00**, length 2 contains the packet data: in this case 2 bytes, values 0xaa and 0xee. Field **0x83** denotes end of packet containing destination address (**0x03** with highest bit set). The address type is 802.15.4 long address (8 bytes, indicated by the first byte after address length field **0x01**), with value **0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF**.

Example configuration packet:

Table 10: nRP example nRoute configuration packet.

Field tag	Length	Value	Field tag	Length	Value
N/A	4	4e 52 50 00	**0x00**	1	00
0x01	9	01 FFFFFFFF FFFFFFFF	**0x83**	2	FE

This nRoute configuration packet will instruct the receiving nRouted to forward all data packets that have protocol **0x00** (IEEE 802.15.4 MAC data), originating from any address (**FF:FF:FF:FF:FF:FF:FF:FF** being a broadcast address), port **0xFE** (254, nUDP nPing port).

If the previous configuration packet was parsed successfully by the nRouted, it will reply with a following packet.

Table 11: nRP example nRoute configuration packet.

Field tag	Length	Value	Field tag	Length	Value
N/A	4	4e 52 50 00	**0x05**	2	“OK”

