Radiocrafts
Embedded Wireless Solutions

User Manual

RIIM™ – Radiocrafts Industrial IP Mesh
Rev. 3.0.0

Table of Tables

Table of Figures

Table of Examples

Abbreviations

Abbreviation	Description
ADC	Analog-to-Digital Converter
AFA	Adaptive Frequency Agility
API	Application Programming Interface
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
GPIO	General Purpose Input/Output
GW	Gateway
I^2C	Inter-Integrated Circuit
ICI	Intelligent C-Programmable Interface
LBT	Listen Before Talk
MAC	Media Access Control
OSI	Open Systems Interconnection
PAN	Personal Area Network
PHY	Physical Layer of the OSI model
RF	Radio Frequency
RIIM	Radiocrafts Industrial IP Mesh
SDK	Software Development Kit
SLIP	Serial Line Internet Protocol – IPv6 over UART in RIIM
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol

1 Overview

Radiocrafts Industrial IP Mesh (RIIM™) is a wireless sensor network that can send and receive data directly on the internet. This makes it easy to connect to cloud services and other devices. IP addressing is used to address each module from anywhere and vice versa. The IP-Mesh system consists of the RC1882-IPM module. This module is versatile and can be used as network Border Router, Mesh Router node and low-power Leaf node. Radiocrafts also provide a finished deployable gateway, the RIIM Border Router – See www.radiocrafts.com for details on the deployable RIIM Border Router and an overview of the RIIM ecosystem.

Sensors can be connected to the RC18x2(HP)-IPM using common interfaces such as I2C and SPI, and the user can interface them through an easy-to-use programming API named ICI (Intelligent C-Programmable Interface). For ICI application development, Radiocrafts provide a complete SDK (Software Development Kit) with firmware, tools, documentation, and examples for sensors to minimize the work needed for development. The API also provides the possibility for the user to develop low-level custom drivers and can be used to implement arbitrary functionality. The modules are inherently low power modules with automatic power save functions enabling battery operation.

2 Introduction to RIIM

The RIIM Network consists of the following key elements

- The RIIM SDK
  - Software development kit with the ICI application framework and tools for creating and uploading end ICI applications to the RC18x2(HP)-IPM
- The RC1882-IPM module
  - The RC1882-IPM module can be configured as a Border Router node, Mesh Router node or Leaf node.
    - As a Border Router it acts as the base of the mesh network. It can connect to an external network via ethernet or custom user ICI application on other interfaces such as UART
    - As a Mesh Router, it relays packets in the RIIM mesh network
    - As a Leaf, it does not relay packets to other nodes except its parent. This mode uses the least amount of energy.
  - All node configurations require an ICI application for RF and interface configuration and the user application. The same RIIM Software Development Kit (SDK) is used to create the ICI application for all node configurations.

Below is an illustration of the different elements and the documentation available

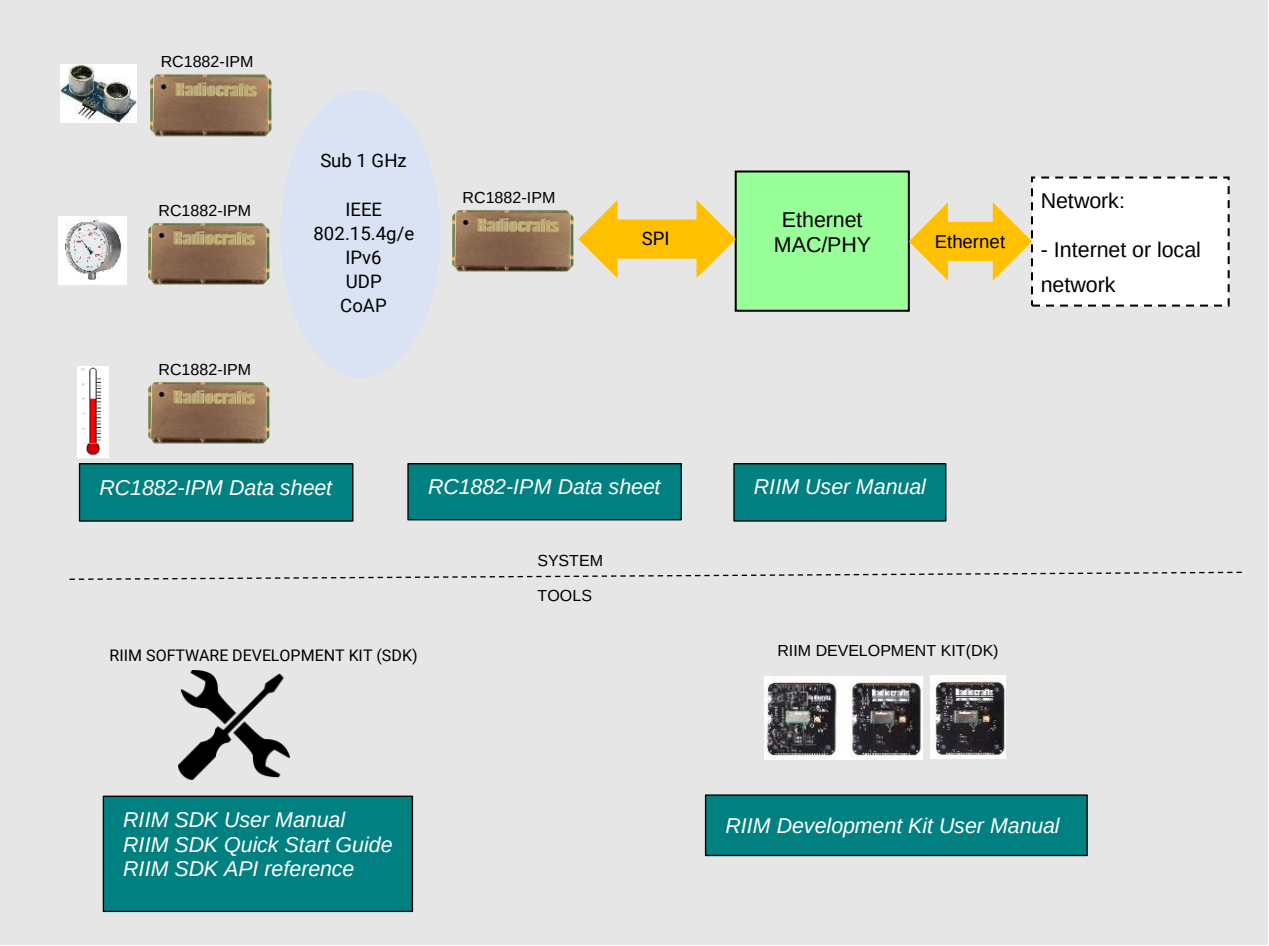


Figure 1. RIIM Network – system and documentation overview

3 RIIM Mesh Network

The following figure shows a typical Wireless Mesh Network. The network is self-forming and self-healing, so the user does not need to configure anything; the network will form itself. The network is a tree structure, where one node (the Border Router) is the root node, and the others (child nodes) are branching out from the root. Multihop is supported, so it is not necessary for a mesh router or a leaf node to have a direct link to the Border Router.

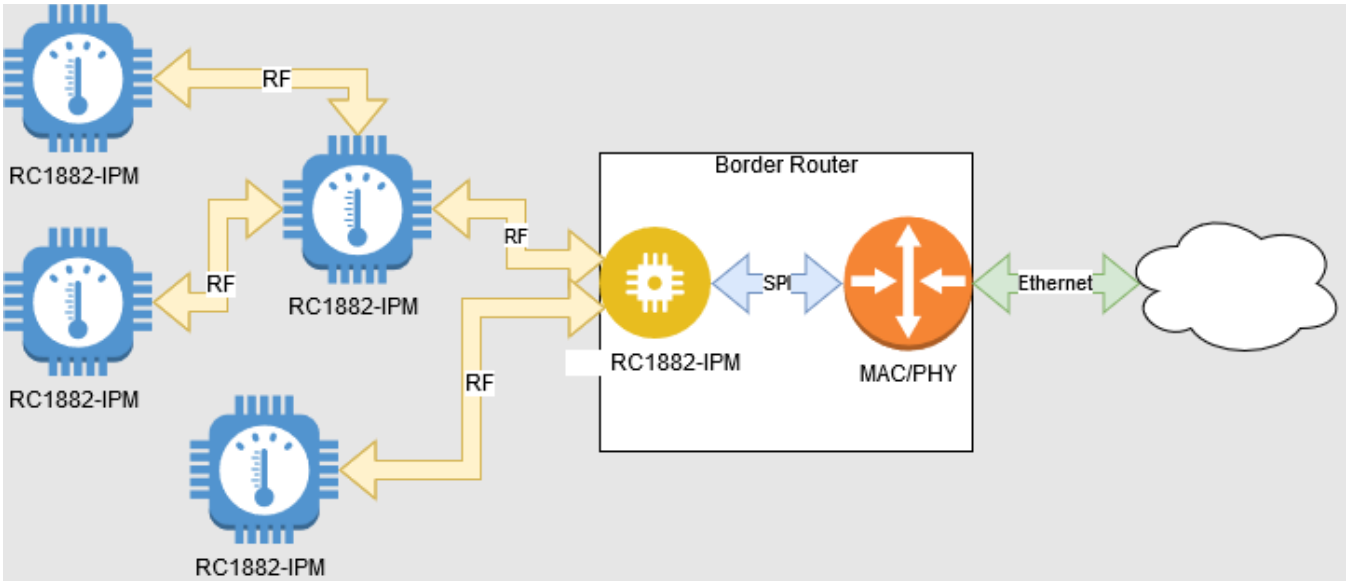


Figure 2. Example IP Mesh network

3.1. Overview

A complete deployed solution will consist of at least one Border Router node. The rest of the RIIM network will consist of Mesh Router nodes and Leaf nodes (collectively referred to as c*hild* nodes). It is currently not possible to use multiple Border Router nodes in the same network. However, it is possible to have a backup border router and a contingency plan in case the border router fails. This means that though ICI one can verify connection to border router and if this connection is missing for a period, the backup border router can be used. This eliminates single point of failure.

3.2. RIIM Selection guide

A RIIM solution has many technical advanced features. However for a given use case normally one or two features are the key factors.Trying to use all the features simulataniously might not give an end results as expected. Therefore there are a couple of question that needs to be considered for the specific use case and the answer will guide the final solution.

Q&A

- What is the range between nodes?
  - This question is related to using high power modules (27 dBm) or low power modules.
  - How long distance the wireless nodes can handle depend on background noise and environment between nodes.
  - As a guideline for network planning, low power modules shall be able to handle 30 meters indoor through several walls and 500 meter open area outdoor. For a use case with more challenging links, the high-power modules should be considered.
- What region are you operating?
  - RIIM currently support 868 MHz and 915 MHz. Depending on where in the world the network is operating the local regulation might allow one or two of these bands.
  - 915 MHz band is used for north America and South America + Australia/New Zealand + Vietnam
  - 868 MHz band is used in Europe, but also most of Africa, India and Russia
- How is the solution connected to the internet (if at all connected)?
  - There are several different use cases for internet connectivity.
    - Self-contained network, no connection to internet/cloud
    - Router connection to the internet
    - Edge gateway connection to the internet

    The difference between router connection and edge gateway is that a router forwards the packets on IP level without caring about the content, while an edge gateway includes intelligence and edge computing.
    Also, the use case must define if the physical connection is based on ethernet, Wi-Fi, 4G or other. See Table 1 Options for network connection and chapter on cloud connection.
- How many nodes in one network?
  - RIIM can support up to 1000 nodes in one network. However, the throughput per device will be lowered and the extension up to 1000 must be evaluated vs other network performance.
- What does the data traffic pattern look like?
  - In a wireless mesh the traffic pattern is important to understand.
    - Is the data mostly from sensor to a cloud or to a gateway
    - Is there communication between nodes in the network
      - In some use cases a node shall only send data to nearby nodes. This is solved in RIIM with one-hop multicast.
    - Is there downlink data from the gateway/cloud to any device

    Based on a traffic analyses the network operator must consider the following:
    - What is network traffic for the border router and mesh routers close to border router.
    - Too many packets to/from one device could lead to network congestion. Typical 3 packets per second is a safe limit. Increasing above this must be done with care and if the network is optimized for low power consumption the limit will be lower.
    - For high traffic devices the TSCH with AFA is important to fulfill EU radio regulations
- Are devices in the network battery operated?
  - There are 3 options to consider:
    - All nodes have power (e.g. smart street lighting or smart metering for electricity meters)

- There is a backbone of mains operated device in combination with battery operated sensors.
- All nodes in the network are battery operated.

For the first option any variant of the RIIM can be used.
The second option is best solved with single channel operation, which include sleeping leaf nodes with down to 4.7 uA current consumption.
For the last use case a battery operated mesh(sleepy mesh) based on TSCH is required. For the two first use cases a single channel option is best.

- Latency requirement
  - What is the timing requirement and the wireless data?
    If the data upward toward border router is important then a single channel system is best
    If the data downward from border router to mains powered nodes are time critical then a single channel system is the best solution
    If the data downward from border router to battery operated nodes are time critical, then TSCH is the best solution.
- Network topology
  - Network topology will also have an impact on how the network preforms. A long and narrow network will have many hops and behave differently than a shallow and wide network. Network congestion analysis will be different for different topologies
- Time synchronous events
  - Do the use cases required that node behave in a synchronous matter? That is that they do action on the same time or is there a requirement that nodes never act in the same moment. All this can be controlled with the time-synchronized events of RIIM. Time synchronized events requires TSCH

Table 1 Options for network connection

	Router		Gateway	
	IPv4	IPv6	IPv4	IPv6
Ethernet	NAT64	SLIP to external GW with Ethernet, 4G or WiFi	NAT64 + LAN connection to GW	SLIP to external GW with Ethernet, 4G or WiFi
4G	NAT64 + LAN to 4G Router		SLIP to GW	
WiFi	NAT64 + LAN connection to WiFi access point			

3.3. RIIM Network Setup

Child nodes joins the network automatically if they are allowed; there is no explicit “join” sequence that must be executed. The following table shows what criteria is used to allow joining of a node into a RIIM Network:

Network Description	Criteria for joining network
No link layer encryption (RIIM platform versions prior to 1.1.0)	• PAN ID must be the same for all nodes • RF Channel must be the same for all nodes
LLSEC enabled (All other RIIM platform versions)	• PAN ID must be the same for all nodes • RF Channel must be the same for all nodes if using Single Channel • Network key must be the same for all nodes

Table 2. Criteria for joining a RIIM network

3.4. Border Router Connection

The border router has 3 main functions:

1. Function as a regular node with support for ICI applications
2. Function as the Border Router node for the RIIM Network.
3. Function as the connection to an external network

1 and 2 is implicitly supported and autonomous. Connection to an external network will require some configuration. If you are using the Radiocrafts RIIM Border Router, Ethernet works out-of-the box. If you are to develop your own regular Border Router based on the RC1882-IPM module, and want to use Ethernet, you must connect the SPI bus to a Microchip ENC28j60 which is natively supported by the module.
If you are using your own board without ENC28j60 (Ethernet/LAN connector), you must use the Standalone Border Router platform.

3.4.1. SLIP

The Standalone Border Router platform supports SLIP, which is basically IPv6 over serial line (UART). This can be used to connect the Border Router to an SBC, for instance a Raspberry PI, and have full IPv6 addressing capabilities. Radiocrafts provides example setup for this.

3.5. External Node Addressing using Ethernet and IPv4

Each node is assigned a port number on the Border Router Ethernet connection. For the Border Router, this port number is always 10000. The child nodes are assigned port numbers in increasing order according to their index in the network topology. So, child node 1 is assigned port number 10001, child node 2 is assigned port number 10002 and so on. To retrieve the indexes, you must read out the topology. See chapter “Get the RIIM Network topology”.

3.6. Frequency and output power

The RIIM mesh network is available in two frequency bands:

- 863 - 870 MHz for Europe (referred to as the 868 MHz band)
- 865.125 - 866.725 MHz for India
- 902 - 928 MHz for FCC (US) (referred to as the 915 MHz band)
- 915 - 928 MHz for Australia (referred to as the 920 MHz band)
- 918.8 - 921.8 MHz for Vietnam

Future variants may also include 433 MHz and 2.4 GHz.

All bands support 50 kb/s data rate. For single channel mode it is recommended to use the 868 band and the 915 MHz band. These support 34 channels in the 868 MHz band and 129 channels in the 915 MHz band. The channel umber corresponds to the numbering in IEEE802.15.5g.

The dedicated bands for India, Australia/New Zealand and Vietnam should be used in TSCH mode.

Number of channels used for frequency hopping is shown below:

	TSCH mode	
	Number of hopping channels	AFA (see ch 5.2.5)
868 MHz band(EU/CE)	16	Yes
915 MHz band(US/FCC)	50	No
920 MHz band(AU/NZ)	20	No
865-868 MHz (India)	9	Yes
918-921 MHz (Vietnam)	16	Yes

For both frequency band high power modules with up to 27 dBm are available – RC1882HP-IPM and RC1892HP-IPM. These give extended range due to higher transmitted output power, better sensitivity and better interference robustness (e.g. vs. 4G/5G based stations) with SAW filter.

There are two MACs to choose from, single channel (CSMA) and time-slotted multiple channel frequency hopping (TSCH). TSCH utilize 16 channels in the 868 MHz band, 50 channels in the 915 MHz band and 20 channels in the 920 MHz band. See chapter 5 for detailed information about CSMA and TSCH.

There are regulatory limitations to the use of these module and which MAC modes are used. The table below gives an overview

	868 MHz band	915 MHz band
Single channel, output power 14 dBm	All 34 channels possible	Output power must be adjusted to below -1 dBm for this to be FCC compliant (§15.249)
Single channel, output power 27 dBm	Only channel 32 (869.525 MHz)	
Frequency hopping (TSCH), output power 14 dBm	Hops on 14 of 16 channels(Adaptive Frequency Agility remove the 2 worst channels)	Hops in 50 channels to be compliant to FCC §15.247
Frequency hopping (TSCH), output power 27 dBm	Not applicable	

See chapter 5.3 for more details on single channel vs TSCH frequency hopping.

3.7. Connection to the Internet and Cloud/Server Solutions

As all nodes have their own IPv6 address, they can be a part of the internet just like any other internet connected device. Basically, they can address anything, as well as being addressed by anything.
Key network (external side/ethernet) parameters are listed in the following table:

Parameter	Value
Application layer protocol	CoAP (without DTLS) or CoAPs (with DTLS)
Transport protocol	UDP
Transport layer ports	Depending on which node to access. See chapter on node addressing.
IP addressing	IPv4 externally using Ethernet. Automatically using DHCP if available
Internet protocols	IPv4 (Ethernet). IPv6 (Internally and SLIP)
Link MAC/PHY	• 10BASE-T Ethernet when using RIIM Border Router and Microchip ENC28j60 based networking • UART when using SLIP

Table 3. Key network parameters

However, issues with IPv4 translation, firewalls, UDP connections, filters etc. can introduce problems to the flow of data. The most common problems and solutions are described later in this document.

The figure below shows a typical setup for a RIIM deployment.

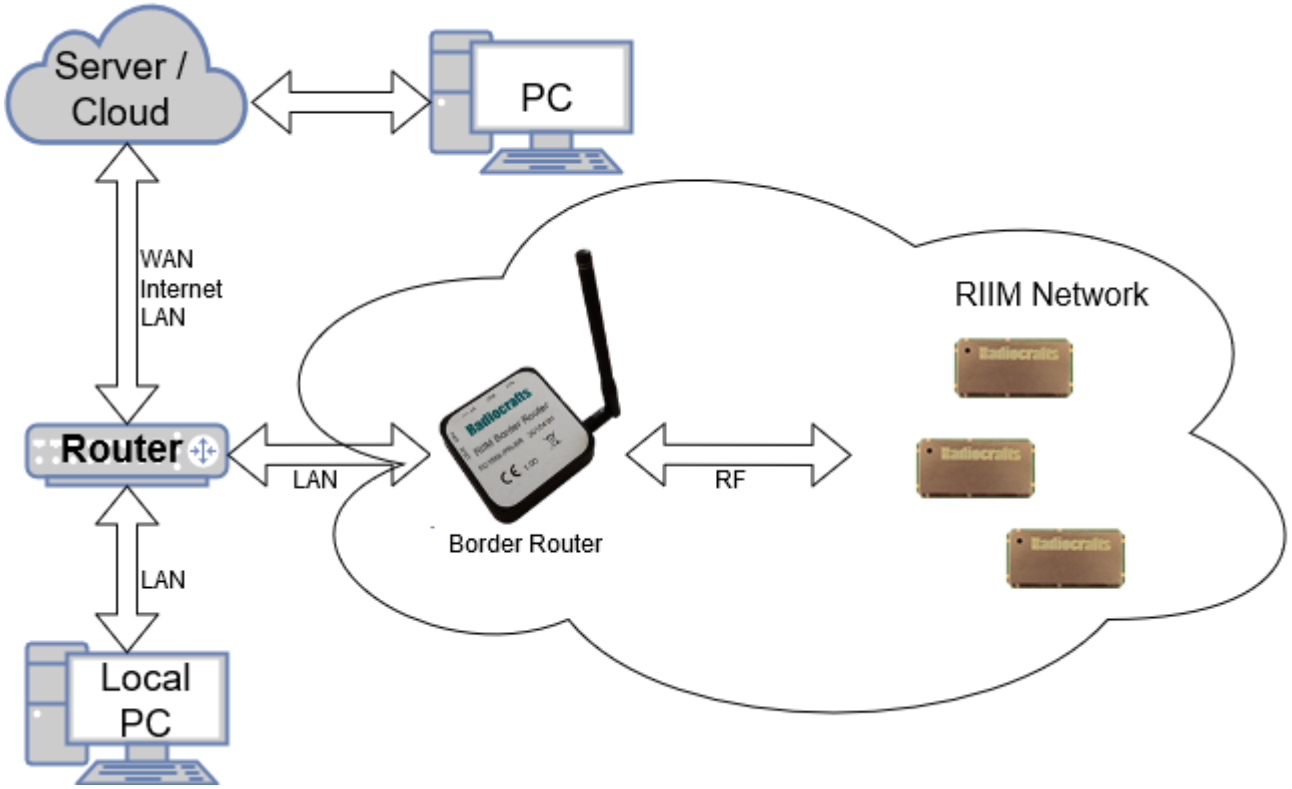


Figure 3. A typical RIIM deployment

In this example, all the nodes in the RIIM Network (including the Border Router) needs to be accessible from the server/cloud, and vice versa. The obvious problem is that the RIIM Network does not know the IP address of the server, and the server does not know the IP address of the RIIM Network. The task of exchanging addresses might be non-trivial, and the following list describes some of the problems and solutions:

- Depending on the firewall policies, “stray” UDP packets from outside the LAN may be stopped by the router, which means that the exchange of addresses cannot be initiated from the outside cloud/server. Possible solutions:
  - Initiate the exchange from inside the LAN
  - Configure the router/firewall
- Router may not keep traffic open for incoming UDP packets forever. Possible solutions:
  - Nodes may need to periodically send packets to server and vice versa to keep channels open
  - Configure the router/firewall
- The address of the Border Router is unknown after it has been connected to the LAN. Possible solutions:
  - The address of the Border Router can be dumped on the UART if the ICI application calls the “Debug.printSetup()”-function
  - The address may be extracted from one of the address tables in the LAN router, depending on the router type and setup.
  - Using ICI, the Border Router can itself send a message to a known address upon power up. The address can be hardcoded in the ICI application or stored in non-volatile memory.
  - When using SLIP, the Linux TUN device itself can be used for commissioning. Radiocrafts provide such an example on GitHub.

There are several ways to exchange the addresses between the server and the RIIM Network given the above points. The following chapters discusses some of them.

3.7.1. Hardcoding the server address using ICI

If the address of the server is known, it can be hardcoded into the module in the ICI application. Beware that the server address cannot change permanently unless you update the ICI application. However, firmware upgrade is easy through OTA. See the SDK documentation for details on setting the server address.

3.7.2. Custom ICI sequence

The ICI application gives virtually infinite possibilities regarding custom commissioning sequences. Provided for the ICI application is also non-volatile memory that can be used by the application to store/update the server address(es).

3.7.3. Using a PC on the LAN

Using a local PC to provide the RIIM Network with the address to the cloud/server ensures that the address exchange is initiated from the LAN. This works the following way:

1. The Border Router node is provided with the address to a server and the name of a CoAP resource on that server.
2. The Border Router node sends a CoAP message to the server with some identification. This can for instance be the IP address.
3. When the server sends its ACK response, the sequence is complete. The server now has the node IP and the node has the server address.

3.7.4. Accessing RIIM Network nodes directly from the internet or LAN

Even if the address of a child node is known, some prerequisites must be met to allow direct accessing. The main problem arises as a translation is needed between the LAN IPv4 to the IPv6 used in the RIIM Network. To allow direct access, the Border Router implements a NAT which assigns a port number to a RIIM Network node. The NAT in RIIM platforms prior to 1.1.0 is dynamic and is initially empty. On later versions, port numbers are pre-assigned to child nodes as described in the chapter “Node Addressing”. There are several ways to “fix” this problem:

- To fill in the NAT table, a child node will first have to send a packet out onto the LAN. When this happens, a port is assigned to the node, and it can be accessed using the Border Router IP address and its assigned port. The Border Router supports up to 32 custom NAT entries at any one time. This is no issue in RIIM platforms version 1.1.0 and later.
- A packet coming from outside can be sent to the Border Router instead of directly to a RIIM Network child node. The Border Router could then run an ICI application that resends the packet to the correct node. For instance, part of the payload sent to the Border Router could be the IPv6 address of the node. Or it could be an index.

3.8. Border Router Node External IP Address Assignment

The Border Router supports assignment of an IPv4 address to the Border Router node. This makes it accessible to the local network via ethernet. The address is set when the user starts the Border Router, as exemplified in the following code snippet:

Example: ICI code

```
const uint8_t ipv4_address[4]={0,0,0,0};
const uint8_t ipv4_netmask[4]={255,255,255,0};
const uint8_t ipv4_gateway[4]={192,168,1,100};
```

```
RIIM_SETUP()
{
    Network.startBorderRouter(NULL, ipv4_address, ipv4_netmask, ipv4_gateway);

    return UAPI_OK;
}
```

Example 1. Setup of Border Router with a DHCP provided IPv4 address

NOTE: If DHCP is enabled on the network the IP address given by the DHCP server is used

3.9. RIIM Network Address Assignment

IP addresses for the Child nodes are automatically managed. The IPv6 (global) prefix (first 64 bits) is the same as for the Border Router node, and the IID (last 64 bits) is created based on the IEEE address (EUI64) in the module itself. So, for the Child nodes, you cannot change their IP addresses. However, you can read it out, and all Child nodes also knows the address of the Border Router node.

3.10. Security

Security is a key element in any wireless network. RIIM offers two levels of encryption and authentication.

In the wireless network there is a link-layer security that encrypts each packet at the MAC layer. This ensures that only nodes with the correct link-layer encryption key can join the network or listen to traffic in the network. This link layer security is also used for network management packets, so an eavesdropper cannot decode management traffic.
Link layer security use a shared key for the entire wireless network and is therefore often referred to as network security.

Link layer security encrypts the data within the mesh network, but not outside. In order to protect data that is sent or received from the internet, RIIM provides support for DTLS, a security protocol which is part of the internet protocol suite made by IETF https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security.

DTLS ensure end-to-end UDP packet encryption. Through DTLS there is a unique application security key for each device. Since DTLS include negotiation between the end points, DTLS cannot be combined with multicast.

3.11. Multicast

Multicast is an efficient way to broadcast data to many or all devices. The multicast in RIIM is based on the IETF standard MPL, which include both flooding the network with retransmissions and a control protocol for syncing between neighbors and requesting missing multicast packets. This broadcast technique is very robust compared to a simple fire-and-forget strategy.

3.11.1. One-hop Multicast

RIIM also provides a local multicast called link-local or one-hop multicast. This sends a multicast packet to all nodes within listening range but does not traverse the network. This is useful if only nearby nodes need to get the packet and has the benefit of reducing network traffic.

A typical use case for the one-hop multicast is that one node in the network detect an event that the other nodes in the neighbourhood should be aware of. E.g. park-lights that detect a jogger and wants to tell nearby lights to turn on.

3.12. Out of the box solution

For complete out-of-the-box solutions, see our development kit and our stand-alone RIIM Border Router.

3.13. Protocol gateway

Even though RIIM is end-to-end IP capable, one can also add a protocol gateway to enable other transport protocols, as seen in Figure 4. The gateway can convert UDP/DTLS/COAP to TCP/TLS/MQTT(HTTP) and vice versa. Such a solution is more complex and challenging to set up, while it gives the advantage that the two protocol suites can be optimized for different network properties. A protocol gateway can easily be implemented on a standard Linux computer using open source conversion tools.

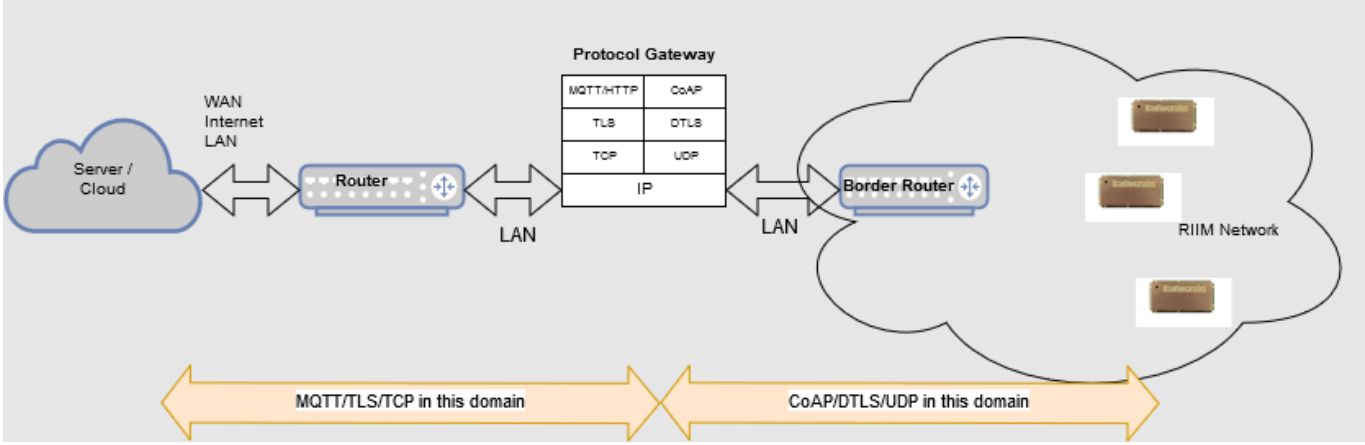


Figure 4. Protocol gateway concept

4 Network Data Packets

The network packets conform to the IEEE802.15.4g standard. They use IPv6 addressing as defined by 6LoWPAN. Every node will have its own unique address, which enables every node to be uniquely addressed from across the internet. The Border Router node provides the translation between 6LoWPAN messages and regular IP messages. No additional Border Router functionality is needed as these messages can be sent directly onto the internet. Some physical connection is of course needed, and the RC1882-IPM already supports ethernet via SPI and Microchip ENC28j60. Additionally, the RC1882-IPM supports a SLIP interface since SDK version 2.0.0.
The transport layer is UDP. It is possible to send packets using only UDP without any application layer and with custom payload. This will save packet overhead and allow for a simpler “fire and forget” way of transmission compared to also using application layer.
CoAP messages using User Datagram Protocol (UDP) can be used to transmit the actual data payload, optionally with encryption using DTLS. Each node can only be connected to one CoAP server at a time for client operations. The node’s CoAP resources can be accessed by multiple clients, as long as not DTLS is used.
CoAP adds to the header size but enables addressing similar to HTTP. It may require a response to ensure delivery of the packet. It is also possible to send a CoAP message without asking for a response, in which case multicast and more power efficient nodes are supported.
Multicast is supported for bare UDP packets and CoAP without response.
Link Layer Security (LLSEC) is implemented and encrypts all packets in the network using the same pre-shared key.

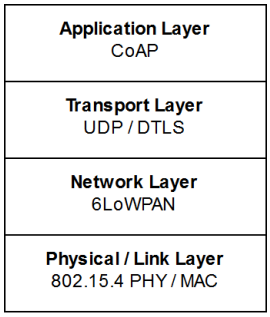


Figure 5. Stack layers

4.1. Data packet overhead

For details on the packet see IEEE 802.15.4 standard.

Packet sent as a regular CoAP request is shown below. Using only UDP will remove the CoAP specific bytes from the header.

SHR		PHR		PSDU					
Preamble	SFD	Reserved	Length	MAC header	MAC Payload				MAC footer FCS
4B	2B	5bit	11bit	9B	15B	7B	11B§	X	2
					IP header	UDP header	CoAP header	CoAP object	

§ The CoAP header includes the resource name used in the URI.

Example:

\\192.168.0.14\log\
192.168.0.14 is an IPv4 address we are sending to.
log is the resource the CoAP request is sent to.
With a resource name of 3 characters the CoAP header becomes 11 bytes. A 12 character resource name gives 20 bytes CoAP header.

The overhead in the above example is 52 bytes.

The IP header in this example sends a packet to its neighbor. Source addressing adds 2 extra bytes per hop, so sending to a node 10 hops away gives 20 bytes extra overhead.

5 Medium access technologies

RIIM comes in two MAC flavours:

- Single channel (CSMA)
  - Single configurable channel operation
  - Asynchronous: nodes can talk at any time
  - Listen-before-talk to minimise collisions
  - Border and mesh routers are always listening on radio

- TSCH (Time slotted channel hopping, announced in release 1.2.0 of SDK)
  - Multi-channel operation
  - Synchronous: nodes are globally synchronized and can only talk during assigned slots
  - Increased QoS due to less chance of interference
  - Enables the possibility of battery-operated routers

5.1. Single channel

RIIM implements single channel CSMA mode. In this mode, all communication happens on one channel. This channel is set in ICI using Network.setChannel() and must be the same for all nodes in the network. Unless in sleep or TX mode, the nodes are always listening for incoming packets. Before transmission, a check on whether other traffic on the same channel is ongoing. In that case, transmission is delayed to avoid collisions.

The «always listen» and single frequency nature of single channel makes data latency low and joining of nodes fast. Single channel provides the best data throughput.

5.1.1. Limitations

As it only runs in single channel, it is prone to interference/jamming on that same channel and collisions from other parts of the same network. Sleeping nodes are not able to hear incoming data, and scheduling of wake-up must be done in the ICI application.

5.2. TSCH

RIIM implements TSCH as specified in the IEEE 802.15.4-2015 standard. In TSCH, the nodes form a globally synchronized network. The network broadcast beacons containing time synchronization information to let other nodes synchronize. Child nodes continuously correct their relative clock drift to their parent through timing information embedded in acknowledgement packetsThe way nodes talk to eachother can be seen in Figure 6 - TSCH time schedule. The colours denote the radio channel used in a slot. The airtime is divided into a continuously repeated set of slots and a node is only allowed to communicate during its assigned slot(s). Within a slot there is time to transmit data and receive an acknowledgement from the destination. Communication will happen on different channels for each repetition of the same slot, and therefore if a packet is lost due to RF interference, its retransmission may be more likely to succeed in the next slot repetition since it will be sent on a different channel. TSCH is suited for low power networks in demanding environments; the nodes can sleep most of the time and only wake up during their assigned slots, and RF interference is mitigated through channel hopping.

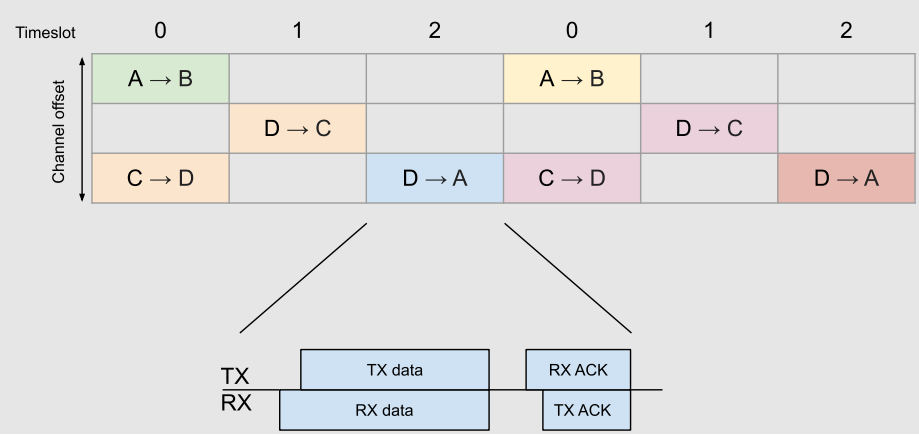


Figure 6 - TSCH time schedule. The colours denote the radio channel used in a slot.

5.2.1. Joining and beacons

The join procedure is a two-step process. First the joining node must synchronize to the network to obtain the time schedule and timeslot information etc. Without this information, the node cannot talk to the network at all. As mentioned earlier, this information is communicated through beacons. The beacons are broadcasted from all nodes on any channel at a user configurable rate. The joining node scan channels at random until it hears a beacon. This crucial first step in the join process, finding a beacon, is the major contributing factor affecting the network join time. How fast the joining node finds a beacon depends on the following:

Beacon broadcast rate

When beacons are transmitted frequently, the joining node is more likely to find one. The broadcast rate is user configurable through ICI and can be set by calling `Network.setTschMaxBrodcastRate(uint16_t rate_s).` Frequent broadcasting will improve join time at the expense of generating more network traffic and increase power consumption. The beacon broadcast rate can be changes dynamically during operation. So to enable smooth joining while keeping low power consumption over time, a small broadcast rate e.g. one second can be set at joining, while a longer beacon broadcast rate can be set when network is stable.

Number of connected nodes within the range of the joining node

Since nodes in the existing network are all transmitting beacons, the joining node is more likely to find a beacon if there are many connected nodes nearby.

The number of radio channels

The beacons are spread on all radio channels, and therefore with many channels the joining node is less likely to find a beacon on the channel it is scanning. The channel count depends on the configured frequency band (also configurable through ICI).See chapter 3.6 for details.

5.2.2. Active or passive mode

When using TSCH, the user can choose between 4 predefined network settings. These are

- TSCH_LOW_POWER
- TSCH_BALANCED
- TSCH_LOW_LATENCY
- TSCH_HIGH_THROUGHPUT_SENSOR_DATA

These all selects parameter values that balance performance vs power usage.
TSCH_HIGH_THROUGHPUT_SENSOR_DATA uses TSCH active mode, while the other settings uses passive mode.

In the active mode the mesh nodes transmit packets at their designated time slot and so it must also listen in slots were it has neighbors of interest. This puts the node in RX mode most of the time and thus have higher current consumption but also higher throughput and potentially lower latency.

In passive mode it is the opposite; the mesh nodes use own their time slot to listen for packets, and other nodes' time slot to transmit packets.. This optimize the battery lifetime of the mesh routers, while sacrificing some latency and maximum network throughput. The passive mode is also referred to as sleepy mesh.

5.2.3. Limitations

Due to the synchronized nature of TSCH and the fact that nodes are only allowed to transmit at certain points in time, the TSCH variant of RIIM has a lower throughput and higher latency than single channel. The network join time is slower because the joining node must catch a beacon containing network synchronization information before it can talk to anyone in the network, as opposed to single channel, where the joining node can initiate the join process. Beacons are scheduled in time slots as other packets and hence there will be a slight decrease in network traffic capacity for application data since they are competing for the same slots.

5.2.4. Listen Before Talk

Before a transmission, the nodes listen to make sure the channel is free to use. This is called Listen Before Talk (LBT). When a node find the channel busy, the transmission is postponed until the next available timeslot. LBT is to prevent collision with external sources, and not within the network itself. This is due to the nodes being tightly synchronized and so two nodes will listen in the exact same period (both thereby considering the channel as available) before transmitting.

5.2.5. Adaptive Frequency Agility

The 868 MHz band integrate a feature where active channels with external interference are replaced with other channels of higher quality. The channel quality statistics is continuously updated and when a channel is deemed bad, it is swapped, and the updated channel list information is distributed to all nodes in the network which adjust their internal channel list accordingly. This feature is called Adaptive Frequency Agility (AFA) and is only activate when the network operates in the 868 MHz band.

5.3. Single channel or TSCH?

Here are some things to consider when choosing between single channel and TSCH.

- Regulatory requirement is a non-negotiable factor. For FCC certification TSCH and frequency hopping is a requirement, while high output power in Europe require single channel usage.
- TSCH enables battery powered routers
- TSCH gives higher reliability as devices transmit on different timeslot and the probability of collisions are reduced.
- Single channel CSMA enables higher throughput and lower latency.
- Single channel CSMA enables very sleepy nodes by disabling them for extended time

6 Network Timing and Latency

6.1. Single channel

The latency in the single channel variant of RIIM is very low, but as with networks in general; non-deterministic. The radio includes listen-before-talk to increase robustness and reduce interference. Packet loss and automatic retransmission will cause extra delay. Following is a method to calculate estimated latency.

The RF data rate is 50 kb/s so 1 byte has an airtime of 0.16 ms.
Based on the packet overhead example given above, and 10 byte CoAP payload, the total RF airtime is 62 bytes x 0.16 ms = 9.92 ms.

The acknowledgment typically comes within 1ms.

The latency in a multihop network is given by

$$LatencyOneWay = TxTime + AckTime + (NumberOfHops - 1) * (RoutingProcessingTime + TxTime + AckTime)$$

To calculate the complete two-way CoAP request and response this latency must be multiplied by 2.
The processing time at the CoAP server end must also be considered.

$$LatencyTwoWay = 2 * \big(TxTime + AckTime + (NumberOfHops - 1) * (RoutingProcessingTime + TxTime + AckTime)\big) + CoapResponseTime$$

The routing processing time is typically 45 ms
A typical CoAP server response is 40 ms.

This will vary depending on what other tasks the device is performing.

Example 1: Sending 10 bytes user data one way to a device 3 hops away

Sending 10 bytes user data to a device 3 hops away.

Latency one way = 9.92ms + 1ms + (3 -1)(45 ms + 9.92ms+1ms) = 122.76 ms

Example 2: Sending 60 bytes user data to a device 1 hop away and get a response back

Latency two way = 2x(16.32ms + 1ms + (1 -1)(45 ms + 9.92ms+1ms))+ 40ms = 79 ms

Effect of packet loss:

Packet loss will create additional delay as packet retransmission must be added. A rule of thumb is that one retransmission of a lost packet typically takes between 40-60 ms. Therefore 50 ms delay is a good estimate.

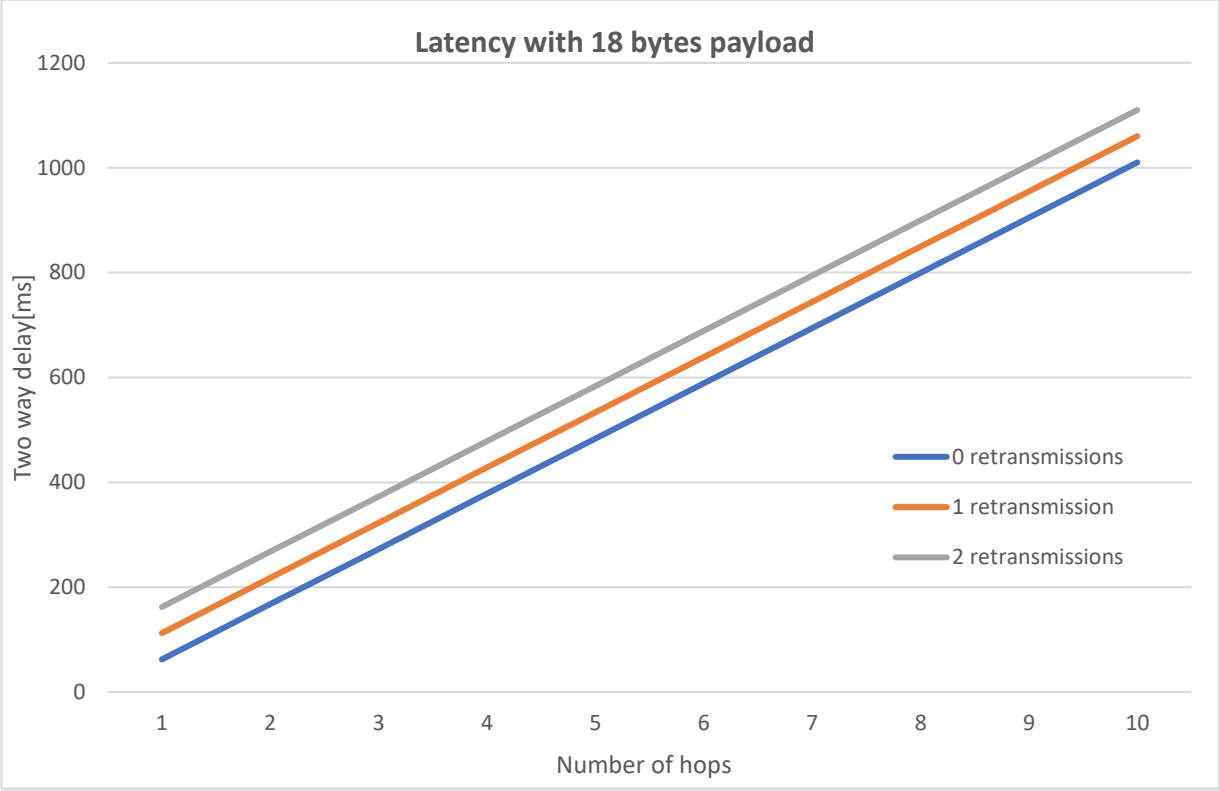


Figure 7. Typical latency for request/response

6.2. TSCH

The TSCH variant of RIIM has a higher latency than single channel since it must wait for the right timeslot before it can transmit or receive. In ideal circumstances, a node waits 340 ms on average for the next timeslot. The length of a timeslot is 40 ms, which includes packet transmission and an optional acknowledgement packet. This is independent of payload size. From this, the average latency in a multihop network is given by

$$
\begin{aligned}
&MeanLatencyOneWay \\
&\quad = TimeUntilNextTimeslot + TimeslotLength + (NumberOfHops - 1) \\
&\quad * (RoutingProcessingTime + TimeUntilNextTimeslot + TimeslotLength) \\
&\quad = (TimeUntilNexTimeslot + TimeslotLength + RoutingProcesingTime) * NumberOfHops \\
&\quad - RoutingProcessingTime
\end{aligned}
$$

From this we can calculate the latency for the examples in 6.1:

Example 1: Sending 10 bytes user data one way to a device 3 hops away

*Latency one way = (340 ms + 40 ms + 45 ms) * 3 - 45 = 1230 ms*

Example 2: Sending 60 bytes user data to a device 1 hop away and get a response back

*Latency two way = 2 * ((340 ms + 40 ms + 45 ms) * 3 - 45 ms) + 40 ms =* 800 ms
This is however under ideal circumstances. A packet might be delayed because of retransmission during packet loss with an exponential back-off feature that will increase the delay significantly, or delay because of other packets with higher priority in the packet queue. This will increase the latency, and actual measurements has

shown latencies of 570 ms per hop with a 97 % link-layer packet delivery rate, and 1055 ms per hop with a 88.5 % link-layer packet delivery rate.

7 Network Throughput

To maximize throughput there are several effects that need to be taken into consideration.

- One by one transmission.
  - If all devices are transmitting at the same time, more interference and higher packet loss will be seen
- Maximize data length to minimize overhead. Sending 20 and 20 bytes is more efficient than sending one and one byte.
- Send next message based on CoAP response. If not the CoAP queue will be filled up and the performance will not be stable.
- If no response is needed, consider using CoAP without response or raw UDP

If all devices in the network are sending data, the border router will be the limiting factor. The simultaneous throughput from hundreds of devices in a large network will in the order of a few bits/second per device.

7.1. Single channel

Based on the above system, the throughput can be estimated based on the latency.

Example 3:
Based on example 2 above the latency with one hop is 79ms/64 bytes.
If we add a processing time of 30 ms for sending the next packet (receiving CoAP response, parse and prepare next packet) we get 64 bytes/116 ms = 4.7 kb/s.

This is based on no packet errors, one hop and one device transmitting, so this is the maximal theoretical throughput.

The theoretical limit for throughput for different number of hops and different CoAP payloads are given in Figure 8.

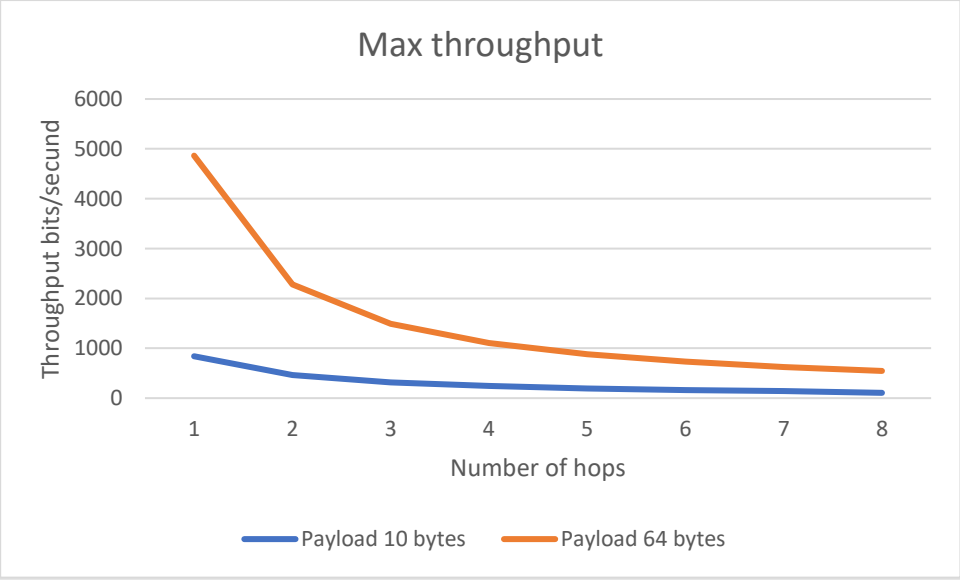


Figure 8. Max throughput

7.2. TSCH

A TSCH node has an opportunity to send a radio packet every 680 ms which, with for exmple a 64 byte payload, gives a maximum throughput of 753 bits per second. The maximum usable payload size per packet depends on things like the protocols used, the number of hops to the destination and if encryption is enabled.

8 Network Congestion

A RIIM network share a common media (RF channels) which is a limited resource. Competing nodes (sending on the same channel, or within the same timeframe) might lose packets due to packet collision.

In a RIIM network packets are stored and forwarded in the mesh routers with finite storage space. Hence with many nodes in a network sending a lot of data there is a possibility of data congestion in the network.
This can be avoided by giving thought to the network and system design. Following are some guidelines for scaling RIIM networks.

8.1. Unicast data to the border router

The Border Router can in theory receive about 30 packets per second (in ideal circumstances, this is unrealistic in real deployments). The real throughput will vary depending on:

- Network topology
- Network density (How many nodes the border router can talk to)
- Single channel CSMA vs. TSCH
- Traffic pattern (N nodes sending P packets vs one node sending N*P packets)

The expected throughput is in the range 3-10 packets per second. This means that a network with 1000 nodes must have a strict limitation on how often data can be sent from each node. In large network it is important to distribute the unicast messages in time. If all 1000 nodes in a large network is “synchronised” with their transmissions, the incoming packet rate will periodically exceed the capacity, resulting in increased packet loss. Therefore, it is important to add random delays for start-up and packet transmission in large networks.

8.2. Unicast data from the border router.

This follows the same limitation as unicast data to the border router. In addition, there is a limitation in the 868 MHz band in Europe for max TX duty cycle that can limit this even more.

8.3. Multicast data from the border router.

Multicast messages traverse the network slower, due to retransmission. The border router can send two or more multicasts directly after one another, but then they will overlap in time when traversing the network. This will increase the probability of packet collision. To avoid local data congestion, multicast packets should be sent with a 5 second interval or more. In addition, the MPL standard requires a pause of at least 5 minutes during or after 255 multicasts to avoid multicast sequence number overflow.

9 Time synchronous events

When using TSCH all nodes in the network are time synchronized.
This can be used to effectively control if nodes in the network shall do simultaneous events or if the application requires event to have fixed offset in time between nodes. The border router acts as the main time source, and all child nodes in the network is therefore synchronized to the border router. The child nodes will automatically compensate for any drift with respect to the border router.
All nodes in the network are aware of a common absolute time, enabling events to happen at the same time, and are also synchronized to within a few milliseconds of each other, enabling near instant action across the network. The absolute time is distributed across the network, and is the same for all nodes. In addition, there are Epoch and Time-of-Day (ToD), that can be calibrated (offset) to the absolute time. See the SDK reference manual for details.

A separate API called the Clock API provides ways to utilize this in an ICI application, and is described n the RIIM SDK User Manual and the RIIM API Reference Manual. The API provides methods for manipulating time, using 24-hour clocks, epoch-based time and enabling timed actions via callbacks.

As the Clock API is dependent on TSCH, it is not available for use when network is configured as Single Channel.

10 Single Channel Node Current Consumption

Current consumption in the RIIM single channel variant will depend on which role the node has in the network and what function it is setup to perform.

Role	*Typical default current consumption*
Border router	9 mA
Mesh Router	9 mA
Sleeping leaf node	4.7 µA

These numbers include the network maintenance functions.

The total current consumption for a sleeping leaf node is depending on what the node activity is:

- How often data is sent
- How often the module wakes up to do other application tasks (e.g. reading a sensor)

For sleeping leaf nodes, the natural way is to implement a client at the sleeping node and a server/resource at the cloud/backend. A CoAP with request/response will keep the sleeping node awake while its waiting for the CoAP response. Thus, the distance to the server and the channel quality are important parameters for the current consumption coming from CoAP request/responses.

Example:

Current consumption CoAP request + response, one hop.

TX current contribution: 27 mA x 67 bytes x 0.16 ms/byte= 268 mA*ms = 290 µA*S
TX MAC ACK contribution = 27 mA x 30 bytes x 0.16 ms/byte = 64 mA*ms = 69 µA*S
RX current contribution = 8 mA x 240ms = 1920 mA*ms = 1920 µA*S

Total = ~2300 uA*S

To find the average current consumption this number must be divided by the seconds between transmissions.

Sending every 5 minutes gives an average current consumption of 2300/300 = 7.7 µA
Sending every hour gives an average current consumption of 2300/3600 = 0.6 uA

The current consumption given by CoAP request/responses are added to the 4.7uA active sleep current to find the average current consumption.

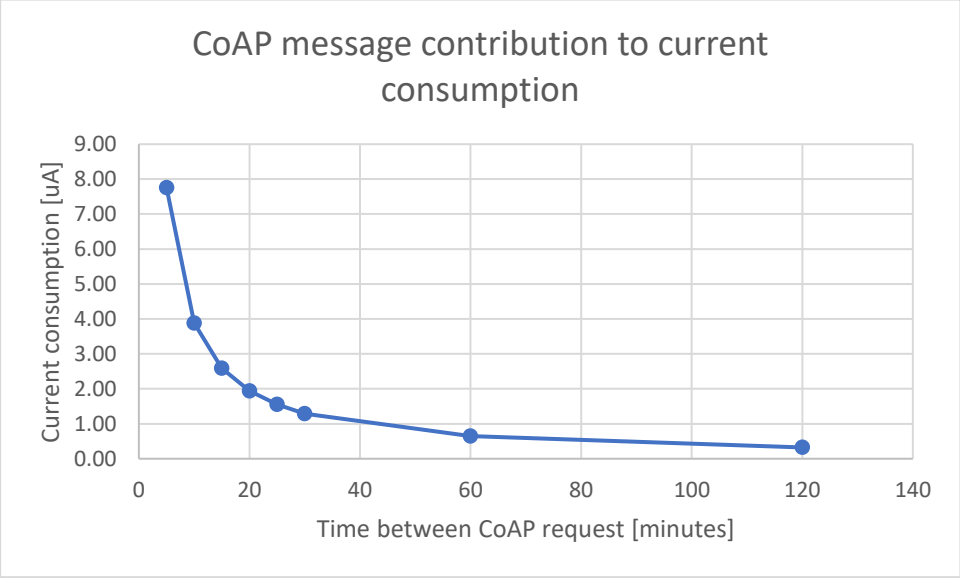


Figure 9. CoAP request/response contribution to current consumption for sleeping leaf nodes

11 Link/transmission robustness

Sometimes, it might be better to ignore nodes with a low RSSI. There can be several reasons for this:

- It can be better to insert a new mesh router than having a weak and unstable link. The link could also be influenced by external dynamic factors, for example a car parking between the nodes
- In dense networks, many nodes can hear each other. This can lead to congestion and fill up the nodes' memory. This, in turn, will lead to network instability

Using the `setRobustnessFactor` function in the Network API might help these issues. The following example shows how to ignore all packages with an RSSI less than -80 dBm.

```
Network.setRobustnessFactor(-80);
```

12 Bootloader

The bootloader is a permanent part of the FLASH memory that is executed before anything else. The bootloader's main responsibility is to update the other software (Radiocrafts Platform and the ICI Application). It is also responsible for enforcing encryption and security on the module and the applications. The bootloader is capable of:

- Load the ICI application onto the module via UART
- Load the platform image onto the module via UART
- Lock the module
- Use encryption keys
- Load the ICI application onto the module from the internal FLASH (when using OTA)
- Load the platform image onto the module from the internal FLASH (when using OTA)
- Print module info

See the **RIIM SDK User Manual** for more information.

13 Configuring and Programming the Module

Each module comes pre-programmed with firmware for the network stack, drivers and application framework. Based on this the customer can program the behavior of each device using a simple API. This eliminates the need for an external MCU. For developing the ICI application and programming the module the user is provided with an SDK. This is described in the **RIIM SDK User Manual**. The ICI application is transferred to the module via UART through the built-in bootloader.

14 Connecting Peripherals

RIIM modules provide many standard electrical interfaces that the user can connect peripherals to. The API includes drivers for all of them, and they consist of easy-to-use functions implemented in efficient C code. The figure shows the pinout and the available interfaces:

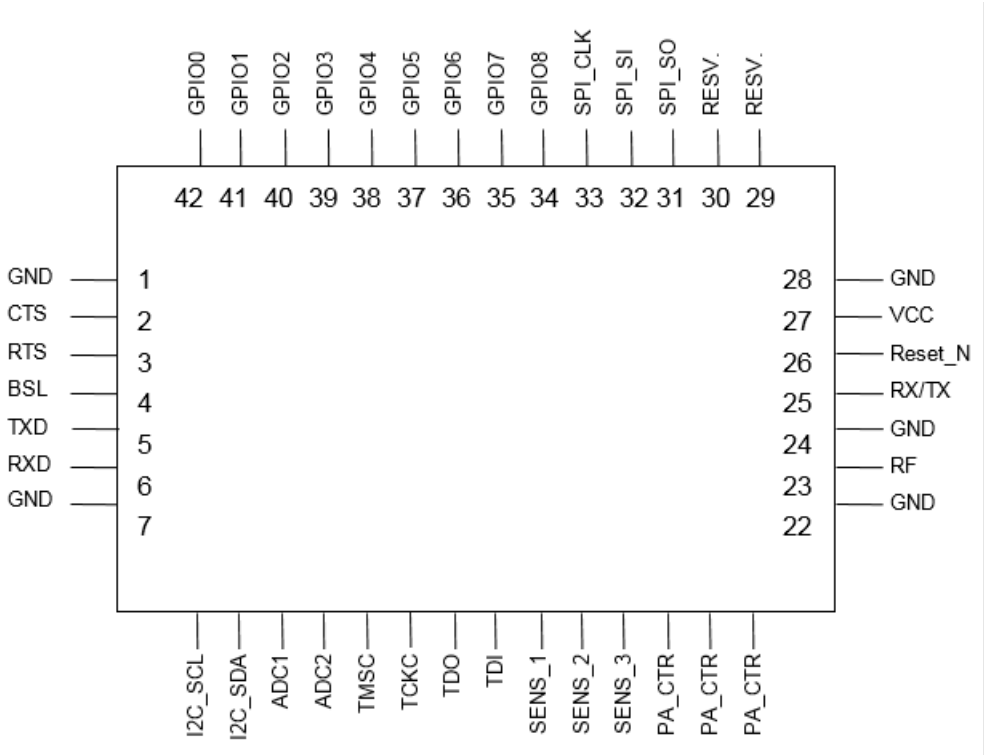


Figure 10. RC1882-IPM module pinout

14.1. GPIO

There are 9 GPIOs available for the user. They can be configured as input or output, with and without pull resistors. They are using the voltage level of the module, and care must be taken to not damage them with higher voltage levels. They can connect to any digitally enabled interface.

14.2. SPI

The module includes one Serial Peripheral Interface bus. All data and clock signals (SCLK, SI, SO) are common to the controller, while the Chip Select that can be implemented in GPIO, enabling the possibility of connecting several SPI devices to the same bus.

14.3. I²C

The Inter-Integrated Circuit (I²C) bus includes internal 4.7 kOhms pull-up resistors and operates at the same voltage level as the module itself. The I²C bus supports both 100 KHz and 400 KHz and supports clock stretching. It operates as master.

14.4. ADC

The analog-to-digital converter converts arbitrary voltages to a digital value. Two analog inputs are provided on the module. In addition, internal channels for temperature and supply voltage are supported.

14.5. UART

The UART provides the classical serial port protocol used by RS232 and serial-to-USB ICs such as many of the FTDI ICs. This enables the user to easily connect to sensors such as GPS modules, or to computer terminal programs. The UART can also be compatible with RS485 by using the CTS pin as RX/TX-pin.

15 Internal module resources

15.1. EEPROM

The RC1882CEF-IPM module includes an I²C EEPROM. This is the Microchip 24AA32A. This EEPROM has the I²C address of 0x50 (0b1010000x). Please see the EEPROM datasheet for details and use the ICI I2C API for interfacing this EEPROM. The EEPROM is exclusively for user data and is not used by any on-board Radiocrafts functions.

16 OTA (Over the Air Download)

OTA is part of the Radiocrafts Platform firmware. When radio packets containing firmware are received, they are stored in the FLASH memory. The firmware is updated after a reset if the entire firmware is received and is error-free. Both the Radiocrafts Platform and the ICI application can be updated using OTA. Security can be ensured by using encrypted images.

17 Border Router Functions

A typical border router is shown below. This is basically how the Radiocrafts RIIM Border Router [1] is built and is all that is needed to realize a complete border router. In fact, you can do without the USB connection as well.

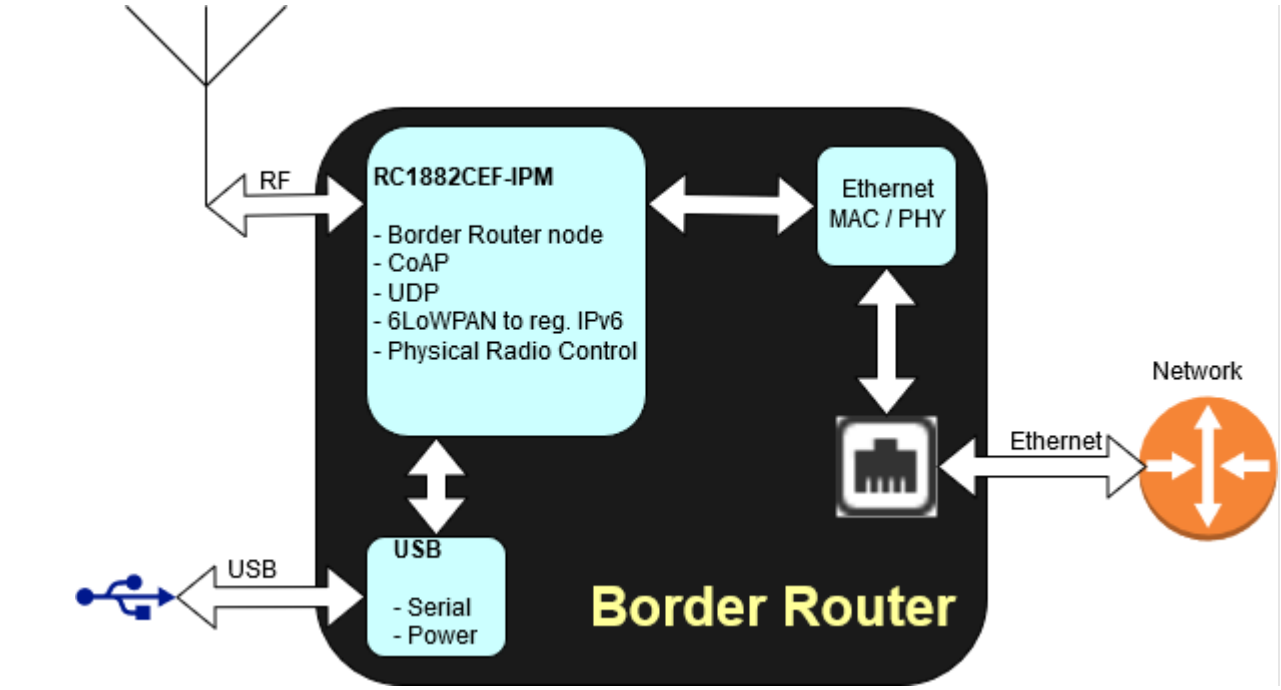


Figure 11. Border Router logical layout

The Border Router uses the RC1882CEF-IPM module to realize all the functionality needed for the Border Router node and the interface towards the local/internet. For easy integration, this is connected to an Ethernet MAC/PHY IC via SPI to the Microchip ENC28j60. As any other RC1882-IPM module, it can run arbitrary code as well. This means that the user can implement their own custom interface to a custom gateway via for instance the UART if that is desirable and simultaneously interface sensors

Figure 12. Radiocrafts off-the-shelf border router

17.1. Programming the Border Router

The Border Router is programmed just as any other node. However, to simply enable the routing, the following program is enough for basic connectivity between the nodes and the Border Router:

Example: ICI code

```
RIIM_SETUP()
{
    Util.printf("Starting RIIM Border Router\n");

    // This example does not use the Ethernet. We provide only NULL
    // as IPv6 and IPv4 configuration
    Network.startBorderRouter(NULL, NULL, NULL, NULL);

    return UAPI_OK;
}
```

Example 2. Basic setup of Border Router node

18 CoAP resources

The RIIM platform includes 3 built-in CoAP resources. In addition, the user can create custom CoAP resources in the ICI application through the CoAP API. CoAP resources are accessed by sending a CoAP message to the URI of the node with the name of the resource at the end, for instance: `coap://[ip-addr]/my_resource`. The resources support GET, PUT, POST and DELETE. All built-in resources support both CoAP and the DTLS secure version CoAPs. All letters in the resource names are lower case.

Generally, Radiocrafts recommends using CBOR (Concise Binary Object Representation, **RFC 7049**) encoding for transmission of data to/from CoAP resources, as CBOR provides a compact, consistent, and easy to decode way of representing data. All built-in resources use CBOR.

Resource	Address
OTA Resource	IP-addr/OTA
Node Resource	IP-addr/Node
Network Resource	IP-addr/Network
Clock Resource	IP-addr/Clock
ICI Application Resource	IP-addr/<User Defined>

18.1. OTA Resource

The OTA resource is a built-in CoAP resource used for over-the-air download of new images.
The overall sequence for transmitting and updating an image is realized with 4 commands:

- Start new OTA transfer (Command 1)
- Transfer image (Command 2)
- Switch image (Command 3)
- Reset node (Command 4)

All commands use the same CBOR structure, and consists of

- Command number
- Address (4 bytes)
- Additional bytes in the payload

18.1.1. Start new OTA transfer

Before starting a transmission of a new image, the FLASH must be cleared. This command can also be used to reset the sequence completely if something goes wrong. The address-field is not used and should be set to zero. There are no additional bytes in the command.

Example payload:
0x83 0x01 0x1A 0x00 0x00 0x00 0x00 0x58 0x00

Decoded, this is:

```
83             # array(3)
   02          # unsigned(2) - Command 1 (Start new OTA transfer)
   1A 00000000 # unsigned(0) - Ignored address field
   58 00       # bytes(0)
               #             - Empty data field
```

Total: 9 bytes

18.1.2. Transfer image

This command transfers a block of the firmware image. The command number is 2. The address specifies where the in the block to start storing, and every image starts at address zero. The additional bytes are the actual block of data

Example payload:
0x83 0x02 0x1A 0x00 0x00 0x00 0x10 0x58 0x08 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

Decoded, this is:

```
83                      # array(3)
   02                   # unsigned(2)   - Command 2
   1A 00000010          # unsigned(16)  - 32 bit address (Address is 0x10)
   58 08                # bytes(8)      - Size of data field (8 bytes)
      1122334455667788  #               - Data bytes
```

18.1.3. Switch image

This command basically arms the update. When entering the bootloader at the start of the execution, the bootloader will look for the presence of a new image. If the image is armed, present and error-free, it will be flashed to the actual program space before execution continues. Address and additional bytes are not used in the command. Module must be reset after this command to do the update.

Example payload:
0x83 0x03 0x1A 0x00 0x00 0x00 0x00 0x58 0x00

Decoded, this is:

```
83             # array(3)
   03          # unsigned(3) - Command 3 (Switch image)
   1A 00000000 # unsigned(0) - Ignored address field
   58 00       # bytes(0)    - No bytes in payload
               #             - Empty data field
```

18.1.4. Reset

This command resets the module

Example payload:
0x83 0x04 0x1A 0x00 0x00 0x00 0x00 0x58 0x00

Decoded, this is:

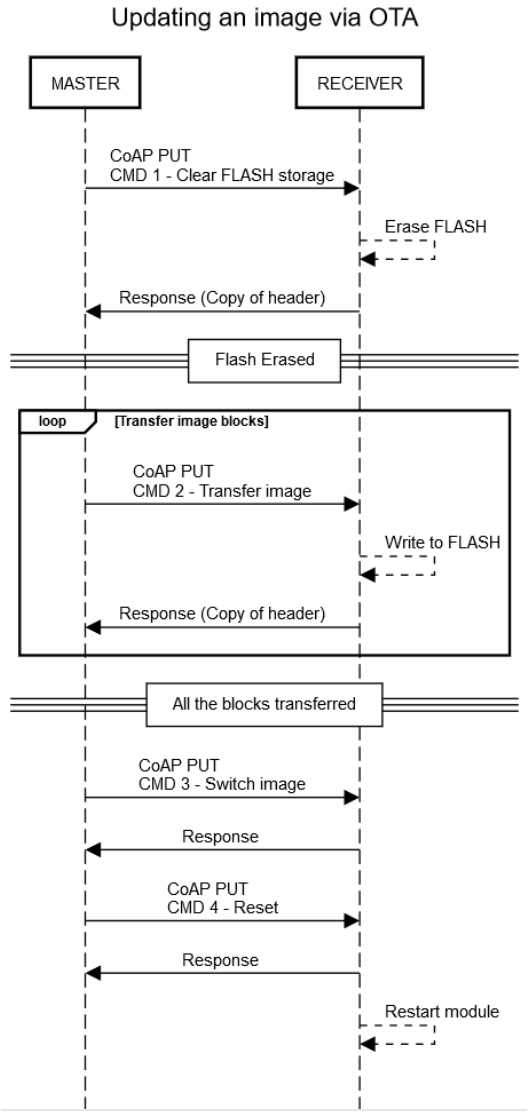
```
83             # array(3)
   04          # unsigned(3) - Command 4 (Reset)
   1A 00000000 # unsigned(0) - Ignored address field
   58 00       # bytes(0)    - No bytes in payload
               #             - Empty data field
```

18.1.5. CoAP OTA response message

All commands get a response with the address of the receiving node plus a copy of the header. The data bytes are not sent in the response. This can be used to verify that a message was indeed received and understood correctly.

18.1.6. OTA Sequence

The following figure shows the whole sequence for a complete OTA



18.2. Node Resource

The Node resource is a built-in CoAP resource used to retrieve information about the node.

18.2.1. GET

Sending a GET message with no payload returns node setup. This table shows the returned payload:

CBOR type	Description
Array of data items	Number of data items in payload. In this case: 7
Tiny unsigned integer	Node type (Border Router, Mesh Router or Leaf)
16-bit unsigned integer	Platform ID
Byte Array of 3 bytes	Platform version (MMmmPP – Major, Minor, Patch)
16-bit unsigned integer	Hardware ID
16-bit unsigned integer	Hardware Revision

Example payload:
0x87 0x01 0x19 0x00 0x11 0x43 0x22 0x33 0x44 0x19 0xAA 0xBB 0x19 0xCC 0xDD

Decoded, this is:

```
85           # array(7)
   01        # unsigned(1)      - Node type
   19 0011   # unsigned(17)     - Platform ID
   43        # bytes(3)
      223344 # 0x223344         - Platform Version
   19 AABB   # unsigned(43707)  - Hardware ID
   19 CCDD   # unsigned(52445)  - Hardware Revision
```

Total: 15 bytes

18.3. Network Resource

The Network resource is a built-in CoAP resource used to retrieve network information. A list of neighbors or the whole network topology can be requested. Not all types of node support all network reporting commands. To request the different reports, the payload of the GET message must contain a command and parameter. These are formatted as CBOR:

CBOR type	Description
Array of data items	Number of data items in payload. In this case: 2
Tiny unsigned integer	Command number
16-bit unsigned integer	Command argument

Command	Command Number	Parameter	Supported by
Get own IP addresses	1	Unused	Border Router, Mesh Router, Leaf
Get neighbour nodes	2	Start index	Border Router, Mesh Router
Get NWS topology	3	Start Index	Border Router

Example payload
0x82 0x01 0x19 0x11 0x22

```
82          # array(2)
   01       # unsigned(1)      - Command number
   19 1122  # unsigned(4386)   - Command argument
```

Total: 5 bytes

18.3.1. Get own IP addresses

This command returns the IP address of the node. It returns 4 different addresses:

- PAN ID
- The local (fe80::)-address
- The global IPv6 address
- The link local address (2 bytes)
- The IPv4 address

This is sent as payload in CBOR encoding. Example:
0x85 0x19 0x12 0x34 0x50 0xFE 0x80 0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0 0x12 0x34 0x56 0x78 0x9A 0xBC 0x50 0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0 0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0 0x42 0x9A 0xBC 0x44 0x11 0x22 0x33 0x44

```
85                                  # array(4)
   19 1234                          #                - PAN ID
   50                               # bytes(16)
      FE80123456789ABCDEF0123456789ABC #                - Local IPv6 Address
   50                               # bytes(16)
      123456789ABCDEF0123456789ABCDEF0 #                - Global IPv6 Address
   42                               # bytes(2)
      9ABC                          #                - Short link local address
   44                               # bytes(4)
      11223344                      #                - IPv4 Address
```

Total: 46 bytes

18.3.2. Get neighbour nodes

This command returns the Interface Identifier (IID) of all neighbors linked to the node and the RSSI of the link to that neighbor. The IID is the rightmost 64 bits (8 byte) of the IPv6 address. The result is taken from a neighbor table in the node, where each neighbor is assigned an index. Get neighbor nodes returns a maximum of 4 nodes each time due to optimization of packet size inside the RIIM Network. This means that the user may need to send multiple GET messages to retrieve all neighbors

0x83 0xE2 0x6C 0x82 0x48 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x42 0xFF 0xAB 0x82 0x48 0x22 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x42 0xFF 0xAB

```
83                          # array(3)
   42                       # bytes(2)
      E26C                  # "\xE2l"      - Responding node's short address
   82                       # array(2)
      48                    # bytes(8)
         1122334455667788   #              - 1st neighbour address
      42                    # bytes(2)
         FFAB               # "\xFF\xAB"   - 1st neighbour RSSI value (2's
complement)
   82
      48                    # bytes(8)
         2222334455667788   #              - 2nd neighbour address
      42                    # bytes(2)
         FFAB               # "\xFF\xAB"   - 2nd neighbour RSSI value (2's
complement)
```

Total: 49 bytes

18.3.3. Get the RIIM Network topology

This request returns the RIIM network topology. All nodes in the whole network and their connections are stored in the Border Router node and nowhere else, so this is only supported by the Border Router. The external connection can transfer larger payloads without packet fragmentation, and a maximum of 2 links are transferred per packet. The links are represented as CBOR pairs of IIDs.

0x82 0x82 0x48 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x48 0x22 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x82 0x48 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x48 0x33 0x22 0x33 0x44 0x55 0x66 0x77 0x88

```
82                          # array(2)
   82                       # array(2)
      48                    # bytes(8)
         1122334455667788   #      - Address 1122334455667788 is connected to...
      48                    # bytes(8)
         2222334455667788   #      - ... 2222334455667788
   82                       # array(2)
      48                    # bytes(8)
         1122334455667788   #      - Address 1122334455667788 is ALSO connected to..
      48                    # bytes(8)
         3322334455667788   #         - ... 3322334455667788
```

Total: 39 bytes.

18.4. Clock Resource

The Clock resource is a built-in CoAP resource used to set and retrieve synchronized time information. There are two values, one Time-of-Day offset, and one Epoch offset. These offsets are with regards to the network absolute time described earlier.

```
83                          # array(3)
   02                       # Type(2) - Offset struct
   1B or 3B                 # Positive or Negative integer
         1122334455667788   #      - Time-of-Day offset
   1B or 3B                 # Positive or Negative integer
         2222334455667788   #      - Epoch offset
```

These are formatted as CBOR:

18.5. Document Revision History

Document Revision	Changes
1.00	Advance Information
1.10	Added commissioning, updated code examples, changed module names
1.20	Added OTA description, DHCP, formatting
1.30	Updated OTA, Commissioning,
1.40	Updated built in resources, intro, added internal EEPROM
1.50	Added timing, throughput, current consumption and radio packet description. Added new 1.1.0 features: UDP, CoAP without acknowledgement, LLSEC, Multicast Restructured the document.
1.60	Added TSCH details
1.70	Added info on frequency bands and High power module with 27 dBm output power.
1.71	Added line about the importance of using Standalone Border Router version when not supporting ENC28j60 on user's own board.
2.00	Updated for RIIM SDK 2.00, with local multicast, AFA, Time syncrounce events, Australia band, Support for high power module, SLIP, support for battery operated mesh routers in TSCH. Q&A section added.
3.00	Updated for RIIM SDK 3.0.0. Described TSCH presets, RS485. Added bands for India and Vietnam

Disclaimer

Radiocrafts AS believes the information contained herein is correct and accurate at the time of this printing. However, Radiocrafts AS reserves the right to make changes to this product without notice. Radiocrafts AS does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Radiocrafts website or by contacting Radiocrafts directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Radiocrafts website. Customers are encouraged to check regularly for the most recent updates on products and support tools.

Trademarks

RIIM™ is a trademark of Radiocrafts AS.

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

Life Support Policy

This Radiocrafts product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Radiocrafts AS customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Radiocrafts AS for any damages resulting from any improper use or sale.

Radiocrafts Webpage

For more info go to our web page : https://radiocrafts.com/
There you can find Knowledge base and Document Library that includes Application notes, Whitepapers, Declaration of Conformity, User Manuals, Data Sheet and more.

Contact Radiocrafts

Sales requests: https://radiocrafts.com/contact/