8 Data Link Layer - Nanolink

This section discusses the Nanolink protocol, which defines the communication between the ground station and the satellte on the data link layer of the radio link.

8.1 Introduction

Nanolink is a reliable, packet oriented, connection based data link layer protocol. The main function of the protocol is to reliably transfer data packets of variable length to another node. To ensure reliability, the protocol uses a combination of forward error correction (FEC) and automatic repeat request (ARQ).

Nanolink is intended to operate in low bandwidth-delay radio links with high asymmetry and moderate to weak signal quality. Reliability and efficiency are achieved by utilizing a hybrid ARQ scheme that is designed to minimize the required reverse channel bandwidth and to be robust to short connection losses. In consequence, higher code rates are possible because the channel coding is not required to handle all errors. Additional losses are handled by retransmissions. Thus, the code rate can be adjusted to achieve maximum efficiency in the average case. The associated higher code rate results in a higher upper bound of the protocol efficiency for best case conditions. Overall, hybrid ARQ can react to changes in the channel conditions more dynamically, since the overhead of redundant transmission only occurs if necessary. For bursty, therefore quickly varying channel conditions, like on the VHF downlink, this is ideal.

Nanolink is the connector between the satellite subsystems and ground control. The protocol manages the physical connection and provides a link for upper-layer protocols. The reliable transfer offered by Nanolink enables the use of simple transport protocols that do not need to implement their own reliable service. This has the advantage that no additional handshakes and transfer of acknowledgments are required. Since it is expected that multiple applications will run on the satellite, this can increase the efficiency of the protocol stack.

8.2 Protocol Features

Transmission Modes Nanolink offers a connection oriented mode and a non-connection oriented mode. The non-connection oriented mode is intended to enable amateur radio operators to receive telemetry data when the satellite is not passing over the LRT ground station. It also serves as satellite discovery mechanism, which is used for connection establishment. The connection oriented mode provides point to point data transfer services between ground station and satellite. This mode is entered only by the initiative of the ground station. The connection is suspended once the physical connection between satellite and ground control breaks and automatically resumed once the satellite's beacon signals are discovered. When a connection is established, transmission modes can be switched on a per-packet basis.

Forward Error Correction To compensate for the frequent impairments on the radio channel, Nanolink employs forward error correction. Since the channel characteristics may change during the operation phase, it is possible to switch between forward error correction schemes, codeword length and code rate. The forward error correction system provides a logical communication channel with lower BER than the actual physical channel. The advantage of this concept is a loose coupling between frames and codewords and the resulting variable frame lengths and codewords.

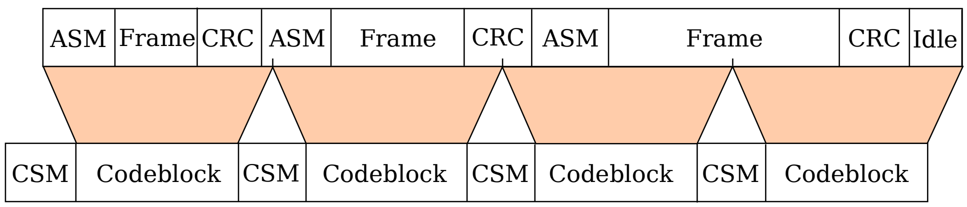

Figure 51: Framing and FEC interaction in Nanolink

Quality of Service Nanolink offers two different types of quality of service: A sequence controlled service that offers reliable delivery and an unreliable service without sequence control.
To ensure reliable transmission, Nanolink combines selective negative acknowledge and cumulative acknowledge.
The unreliable service is thought to be used in cases where reliability is of secondary importance or ensured by other means (e.g. telemetry or timestamp transmission).

Extension Headers The Extension Headers are a distinctive feature of the Nanolink Protocol. They are optional protocol control elements that allow increased versatility and loose coupling without introducing significant constant overhead.
Because the protocol is designed with low bandwidth radio links in mind, an important requirement is low protocol overhead. Not all protocol headers are required in the standard cases and therefore removed from the frame header. Extension header data structures are placed at the beginning of the payload data field. Multiple extension headers may be daisy-chained. New extension headers may be introduced by the user to extend the protocol's capabilities.

Virtual Channels Nanolink offers up to eight different virtual channels to the user. The ARQ state of each virtual channel is independent, which prevents head-of-line blocking of independent traffic. Furthermore, this can be used to implement additional quality of service mechanisms.

Payload The payload data of Nanolink are packets of up to 1021 bytes of binary data. The content of the packet data is not evaluated by Nanolink at any point and therefore ensures loose coupling of protocol logic.

8.3 Terminology

- Byte, An octet of bits
- Frame, data link layer transmission unit
- Protocol Data Unit (PDU), Nanolink Frame
- Service Data Unit (SDU), upper-layer protocol data unit

8.4 Frame Structure

Nanolink PDUs are frames of variable size. Payload and control data are incorporated into a framing structure as illustrated in Figure 52. Each frame is preceded by a 3 Byte Attached Sync Marker (ASM). The ASM numerical value is `0xFAF320`. A frame starts with a 3 Byte header containing information about the contents of the frame. The byte order of the contents of a frame is big endian, 1 byte is exactly 8 bits in length. The frame header is followed by the frame data field, containing optional additional header information and payload data. The maximum length of the frame is 1024 bytes, so that the total maximum data field length is 1021 bytes. Each frame is immediately followed by a CRC-16 checksum that is calculated over the frame.

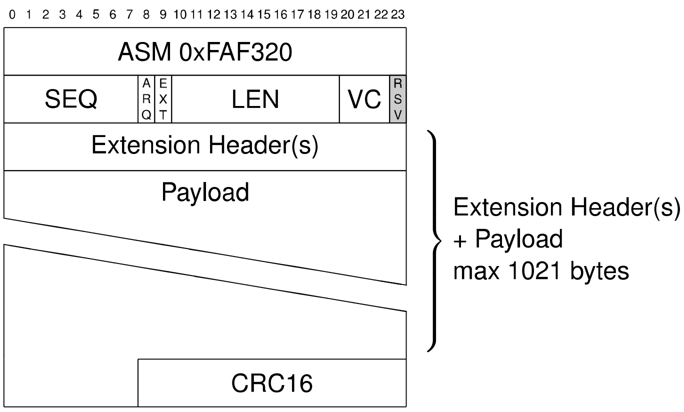

Figure 52: Structure of Nanolink frames

8.4.1 Frame Header

Nanolink PDU begin with the frame header, a 3 Byte long data field comprised of the following fields:

1. The sequence number is an 8 Bit unsigned integer. It is used by the ARQ mechanism to identify individual frames in a sequence, and to detect transmission losses.
2. The ARQ Flag indicates whether or not the frame is sequence controlled and is to be retransmitted in case of transmission losses. If the ARQ Flag is set to '0', the sequence number is ignored by the ARQ mechanism and the frame is deleted after transmission. This mode is used for telemetry beacons, during connection establishment and expendable data.
3. The Extension Header Flag indicates the presence of extension headers in the payload data field. If the Flag is set to '1', the first byte of the payload data field contains an extension header.
4. The length field is a 10 Bit unsigned integer. Its value represents the length of the uncoded PDU excluding the CRC. It is used to delimit the size of the frame and to locate the CRC checksum.
5. The Virtual Channel identifier is a 3 Bit unsigned integer. It specifies the virtual channel this frame belongs to.
6. The last bit of the frame header is reserved for future use.

8.4.2 Frame Content

The payload data field of each frame may contain extension headers, user payload or both. Following the optional extension headers is the user data. This data field contains exactly one SDU of the next layer protocol. Nanolink does neither differentiate between next layer protocols nor indicate the type of SDU in the user data field. It is up to the user to provide suitable means to detect the SDU type.

Extension Headers The very basic extension header structure is displayed in Figure 53. Extension headers may or may not contain payload. This property is dependent on the type of header. Header payload may be up to 256 bytes long.

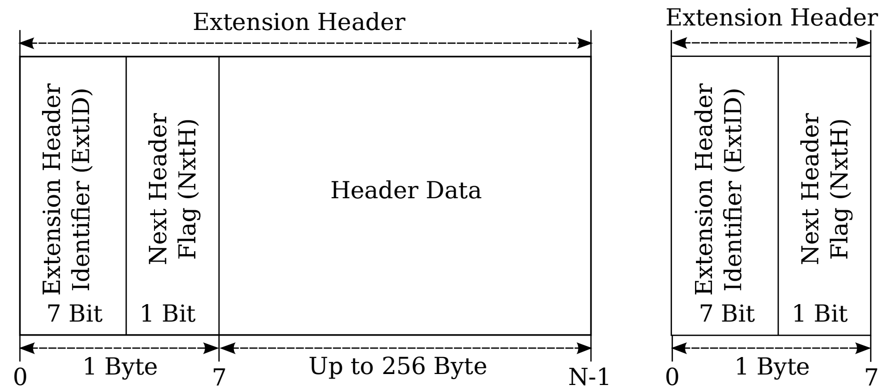

Figure 53: Extension Header Formats: With payload (left), without payload (right)

The extension header packet (see Figure 53) comprises the following fields:

1. The Extension Header ID (ExtID) is a 7 Bit unsigned integer. It indicates the purpose of the extension header and what parameters are to be expected. The length of this field is 7 bits so that the minimal header size is byte aligned.
2. The Next Header Indicator (NxtH) flag indicates if this header is the last header in the daisy chain. If this flag is set to '1', this extension header is followed by another extension header.
3. The header data field is the last extension header field and contains the parameter data. If the parameter size is variable, the first byte represents the length of the parameters. This field is optional and must only be present if the ExtID specifies so.

Extension headers are designed to be daisy chained, so that multiple extension headers can be placed within one frame. The presence of a further extension header is indicated by setting the NxtH flag.

An overview of the predefined extension headers is given in subsection 8.5.

8.4.3 Frame Trailer

Each Nanolink frame is followed by a 16-Bit CRC. The CRC is calculated over the entire frame, except for the ASM and trailer. The CRC follows the CRC-CCITT (nowadays ITU-T) specification, which is:

- Width: 16 bits
- Polynomial: 0x1021
- Initial Value: 0xFFFF

To test the correct implementation of the CRC, the ITU recommends the following test vector:

Test Frame	CRC
C2 A2 15 0D 03 03 02 0B 01	2C 66

8.5 Extension Headers

The following discusses the standard Nanolink extension headers. An overview over these standard extension headers is given in Table 17. The ID represents the integer value of the header, which is inserted in the ExtID field. The length is given in bytes. The maximum number of extension headers per frame is only limited by maximum PDU length. However, of the extension headers listed in Table 17 only one of each ID is allowed per frame. Furthermore, of the ExtID 2–5, only one ID may be present in a frame.

ID	Mnemonic	Length	Description
0	STAT	> 4	ARQ Status Response
1	POLL	1	ARQ Status Request
2	SYN	1	Connection Request
3	SYNACK	1	Connection Acknowledge
4	CC	1	Connection Close
5	CCACK	1	Connection Close Acknowledge
6	PING	2	Test
7	PONG	2	Test
8	CTRLW	User	Media Access Control Word, MOVE specific

Table 17: Overview of Extension Header Types

STAT ARQ status response message. This message contains the state variables of a receiver. Its length is a minimum of 4 bytes. The maximum length depends on the size of the corresponding ARQ window. The structure of the message is displayed in Figure 54.

POLL ARQ status request message. This message contains no body. The structure of the message is displayed in Figure 55.

SYN Connection establishment request. This message contains no body. The structure of the message is displayed in Figure 56.

SYNACK Connection establishment acknowledgment. This message contains no body. The structure of the message is displayed in Figure 57.

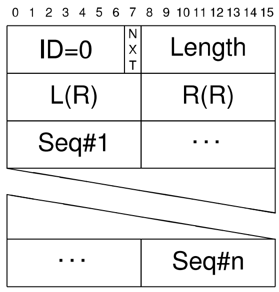

Figure 54: STAT Message Layout

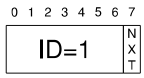

Figure 55: POLL Message

CC Connection teardown request. This message contains no body. The structure of the message is displayed in Figure 58.

CCACK Connection establishment request. This message contains no body. The structure of the message is displayed in Figure 59.

PING Ping message. The message body is an 8 bit integer. The structure of the message is displayed in Figure 60.

PONG Pong message. The message body is an 8 bit integer. The structure of the message is displayed in Figure 61.

CTRLW Media Access Control Word. The message is specific to the application. The structure used in the MOVE-II project is displayed in Figure 65.

8.6 Virtual Channels

Nanolink supports up to eight different virtual channels. Each virtual channel has independent logical queues for new frames (send queue), transmitted frames and received frames. The implementation may vary from this distinction, e.g. merge queues of new and already transmitted frames. Depending on the application, not all virtual channels may be used.

The purpose of virtual channels is to separate independent flows, since the ARQ mechanism may else introduce unnecessary head-of-line blocking. Therefore, each virtual channel provides an independent ARQ state.

Virtual channels may be used to prioritize different flows or applications. This is application and implementation specific, as the receiver is ignorant to this fact.

8.7 ARQ Mechanism

The reliable service ARQ mechanism uses sequence numbers to identify acceptable frames within a sliding window of constant size. Each virtual channel has an independent ARQ state. The sequence number space is defined as the set of all non-negative integers modulo 256. Due to the size of the sequence number space, the maximum window size is 127. Larger windows are not possible since this would lead to wrap-around issues. Sender and receiver share a common window size, so that in total, two ARQ windows exist per Nanolink instance. These window sizes are denoted by W(S) and W(R), for sending and receiving, respectively. The actual window is defined by a set of variables, which are listed in Table 18. These state variables exist for each virtual channel independently. To use ARQ, an active physical connection as well as an active Nanolink connection are required.

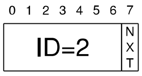

Figure 56: SYN Message

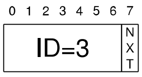

Figure 57: SYNACK Message

In order to move the send window, the sender needs to synchronize its ARQ state with the receiver. Therefore, Transmitter and receiver exchange “STAT(us)” and “POLL” messages. These messages may be sent anytime.

POLL messages are sent by the transmitter for status polling and contain no further information. The receiver responds with a STAT message which contains the highest in-order received sequence number, the highest correctly received sequence number and the sequence numbers within that range, which were not received at the time. The layout of such a STAT message is illustrated in Figure 62. STAT messages can also be sent without prior polling by the transmitter, e.g. after detection of a missing frame. POLL messages may be issued to control the number of unacknowledged frames or to synchronize with the receiver.

8.8 Unreliable Service

The unreliable service is the alternative to ARQ controlled transmissions. It is primarily thought for frames with expendable information, such as control messages, or telemetry data. Secondary users are upper-layer protocols that implement their own ARQ mechanisms. Packets sent via the unreliable service are not buffered by the transmitter and cannot be retransmitted. Unreliable frames have higher priority over reliable frames. The reason for this is that real-time data expires quickly and swiftness is preferred over reliability. This service does not require an active Nanolink connection.

8.9 Procedures

8.9.1 Connections

Nanolink is by design a connection oriented protocol. At the beginning, a connection handshake is performed in order to ensure proper synchronization of the variables of the ARQ mechanism. The connection procedure is divided in an active and a passive part. In the MOVE-II mission, the satellite transceivers assume only the passive part, so that a connection is only initiated by a ground station. This saves resources as no unnecessary connection establishment attempts need to be performed by the transceivers. The connection state of the Nanolink protocol is governed by a state machine. An illustration of the state machine can be found in Figure 63. The machine comprises seven states. States affiliated with the passive part are enclosed in the box marked with “Passive”, states affiliated with the active part are enclosed the box marked with “Active”.

Briefly, the meaning of the states are:

1. Idle – represents the default state in which the device waits for connection requests from a ground station (passive) or for user data (active)
2. Rcv SYN – represents waiting for a valid ARQ frame after a SYN frame has been received
3. Open passive – represents an open connection of the passive part. This is the normal state in which ARQ controlled data is exchanged
4. Rcv CC – represents the state which is entered when a connection close request was received.
5. Send CC – represents sending a connection termination request to the remote Nanolink, or waiting for an acknowledgment of a previous termination request
6. Send SYN – represents sending a connection establishment request to the remote Nanolink, or waiting for an acknowledgment of a previous connection establishment request

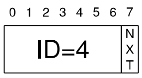

Figure 58: CC Message

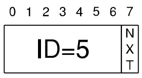

Figure 59: CCACK Message

7. Open Active – represents an open connection of the active part. This is the normal state in which ARQ controlled data is exchanged

States are entered or left by transmission or reception of control messages. These messages are:

1. SYN – represents a connection establishment request message
2. SYNACK – represents the acknowledgment of a connection establishment request
3. ARQ – represents the transmission or reception of a valid ARQ controlled Nanolink frame
4. CC – represents a connection termination request message
5. CCACK – represents an acknowledgment of a connection termination request

The reception of a message is indicated by a preceding “R”, e.g. RSYN. Transitions marked with input ε signify user input, i.e. a user’s wish to establish or terminate a connection. The figure does not contain transitions which stem from protocol errors. In the case of an error, the protocol always enters the *send CC* state in order to terminate the connection.

Connection establishment Prior to an established connection, both nodes are in idle mode. In idle mode, all ARQ state variables of all virtual channels of both nodes are set to the default values, which are listed in Table 19. To establish a connection, the active node sends a SYN message to the passive node. The passive node responds with a SYNACK message for each SYN message. The active node sends the first sequence controlled frame after a SYNACK was received and enters the active connection open state. When the passive node receives a sequence controlled frame, it enters the passive connection open state.

Connection termination Nanolink connections can be terminated gracefully via a connection termination request. However, the expected normal use case is that the connection will be maintained until the physical connection is lost. Therefore, additional timers are used to avoid locking states. Once the Nanolink connection is terminated, the virtual channels are reset to default state.

8.9.2 ARQ Receiver Procedures

The receiver processing of a new frame with number N(R) is based on comparison with the R(R) and L(R) variables and the set of pending frames Q(R).

Window check First, it is determined if a frame is within the receive window, by evaluating if N(R) is in the interval $]L(R), L(R) + W(R)]$. If not, an unsolicited STAT message is issued and the frame is discarded. If the frame is within the window, it is determined if the frame is in sequence to L(R) or R(R), if there has been a loss or if the frame is a retransmission.

In-Sequence If N(R) is directly in sequence of L(R), L(R) is increased by 1. If N(R) is in Q(R) then N(R) is removed from Q(R). If Q(R) is not empty, L(R) is fast-forwarded to $min(Q(R))-1$. If Q(R) is empty and no frames are pending, L(R) is set to be R(R). A STAT message may be issued to signal the sender that no frames are pending anymore. If Q(R) was empty from the beginning, R(R) is set to be L(R).

If N(R) is not directly in sequence to L(R) but to R(R), R(R) is increased by 1.

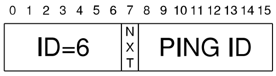

Figure 60: PING Message

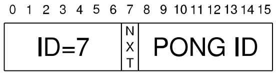

Figure 61: PONG Message

Loss If N(R) is greater than R(R), all sequence numbers between R(R) and N(R) are inserted into Q(R) and R(R) is set to N(R). Depending on the window size and number of lost frames, the receiver may issue an unsolicited STAT to inform the sender of the lost frames.

Retransmission If N(R) is smaller than R(R), the received frame is a retransmission. If N(R) is in Q(R), N(R) is removed from Q(R). If N(R) is not in Q(R), the frame is discareded and an unsolicited STAT is issued, such that the receiver can synchronize its state with the sender.

STAT Transmission If a STAT is requested or required, the message must be included in the next frame of the according virtual channel. If the payload of the next frame is exceeded, a frame containing only the STAT message must be sent instead.

8.9.3 ARQ Sender Procedures

In-Sequence Frames from the send queue are transmitted in the order they were received from the upper layer. New frames may be sent if N(S) is less or equal to $L(S) + W(S)$. If no new frames are available or the send window is exceeded, the sender retransmits unacknowledged frames.

Polling The sender must send a POLL message if the end of the send window is reached or when no more frames are in the sender's queue. The sender should send a POLL message when half of the send window is reached. The POLL message must be included in the next frame of the according virtual channel. If the payload of the next frame is exceeded, a frame containing only the POLL message must be sent instead.

STAT handling Upon reception of a STAT message, the L(S) value is set to L(R) of the STAT message. The V(S) value is set to the R(R) value of the STAT message. All frames with sequence numbers lower than L(S) are deleted from the retransmission queue. If L(R) is equal to R(R), the STAT message contains no further sequence numbers and no retransmissions are required. All frames with sequence numbers which are between L(S) and V(S), but not in the list of the STAT message are deleted from the retransmission queue.

If the STAT message lists missing sequence numbers, the sender begins retransmitting all unacknowledged frames between L(S) and V(S). The retransmission sequence counter is set to L(R). No new frames from the send queue are transmitted until R(S) is equal to the highest unacknowledged sequence number between L(S) and V(S). Should a STAT be received during retransmissions, the retransmission process is restarted using the new L(R), R(R) and R(S). Once all unacknowledged frames have been transmitted, new frames are sent according to the rules of in-sequence transmission.

Special Messages Special messages are messages used for the connection state machine (SYN, SYNACK, CC, CCACK) or the ping/pong functionality (PING, PONG). These messages have precedence over every SDU of any virtual channel and must be sent immediately in an individual frame. Messages of the connection state machine have precedence over ping/pong messages.

Variable	Description
W(S)	Send window size
W(R)	Receive window size
L(R)	Last in-order received sequence number
R(R)	Highest correctly received sequence number
N(R)	Last received sequence number
Q(R)	Set of pending frames in receive window
L(S)	Send window start
N(S)	Next in-sequence number
V(S)	Highest acknowledged sequence number
R(S)	Last retransmitted sequence number

Table 18: Nanolink ARQ State Variables

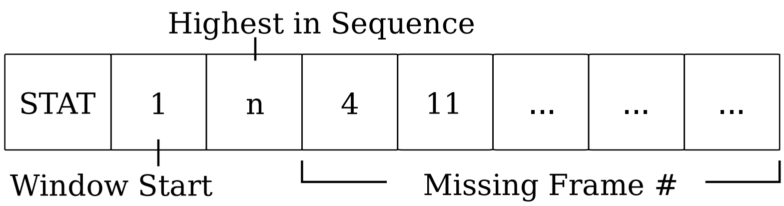

Figure 62: STAT message layout

8.9.4 Ping/Pong

The ping/pong protocol sends minimal messages between two nodes. The sender includes a strictly monotonically increasing number in every ping message. The other node responds with a PONG message, which contains the last received PONG number at the time of transmission. This functionality can be used as keep-alive heartbeat or to measure the round-trip time between two nodes. The protocol is connectionless and does not require an active Nanolink connection.

Variable	Value
L(R)	255
R(R)	255
N(R)	255
Q(R)	∅
L(S)	255
N(S)	0
V(S)	255
R(S)	255

Table 19: Nanolink Default State Variables

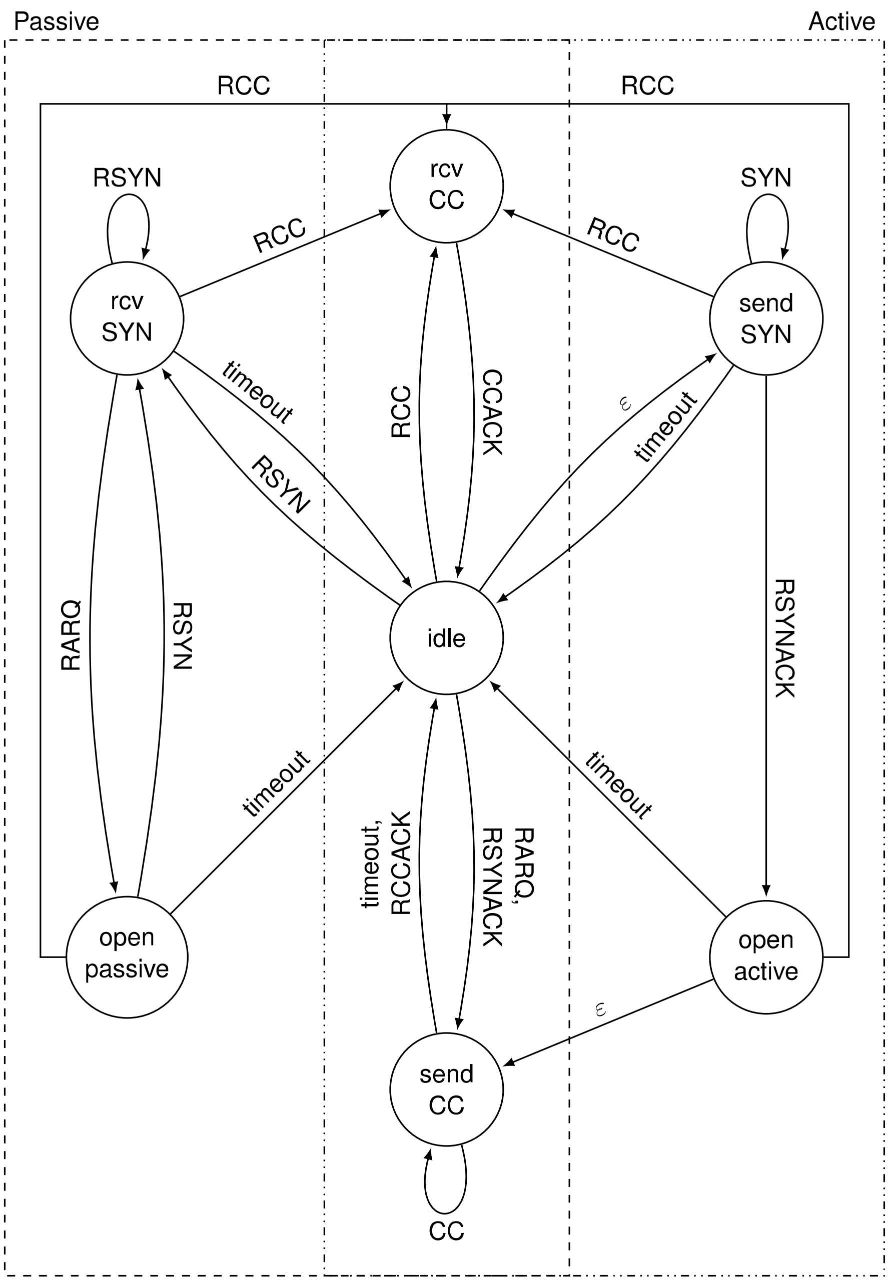

Figure 63: Nanolink Connection State Machine