ClimateTalk 2.0

CT-485 Networking Specification

Document revision: 01
Release: June 12, 2013

Abstract

ClimateTalk is a universal language for innovative, cost-effective solutions that optimize performance, efficiency and home comfort. The ClimateTalk Open Standards define a set of messages and commands to enable interoperability, enhanced user interface, and machine to machine control independent of the physical layer connecting the devices.

This document defines the Networking as well as the Session and Transport requirements for CT-485. Corresponding to OSI Layers 3, 4 and 5, the Networking specification defines the higher-level network functionality including Routing, Addressing, and Network Timing as well as the requirements for a CT-485 device to operate within a defined network structure.

2400 Camino Ramon
Suite 375
San Ramon, CA 94583 USA

This document and the information contained herein are provided on an "AS IS" basis and ClimateTalk Alliance DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITH-OUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. IN NO EVENT WILL CLIMATETALK BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

This document is subject to change without notice.

Updates

This specification may be updated at any time and may be superseded by a more recent version or amended to from time to time. Users should be certain they are using the current ClimateTalk version and the latest revision of the documents.

The released versions of all specifications are available at http://www.ClimateTalk.org

Version History

ClimateTalk Version	Document Revision	Release Date	Comments
V 0.9		2008-11-07	Pre-Release
V 1.0		2009-08-24	Initial Release
V 1.1		2011-06-23	Errata Package
V 1.3		2011-11-02	Additional Errata Updates, Revised Formatting
V 2.0	00	2013-01-18	Version 2.0 Release –Increase number of nodes allowed on the network from 8 to 40. Update send methods to enable multiple nodes of the same type. Improve token offer broadcast & coordinator arbitration to support forward & backward compatibility.

V 2.0	01	2013-06-12	Updated specification to indicate a maximum of 15 different device types (as identified by node type) can be on the network at one time when a CT1.0 device is in the node list (due to CT1.0 backwards compatibility) Updated 6.3.4 Send Method 3: Routing by Socket to clarify node list position refers to the node list sent to CT2.0 devices. Updated 8.1.2.3 Network Node List Example and Framing for CT2.0 devices to include CT2.0 devices addressed on subnet 3. Added 8.1.2.4 "Condensed" Network Node List Example and Framing for CT1.0 devices to ensure CT1.0 devices will detect CT2.0 devices on the network. Updated 11.4 Coordinator Arbitration to explicitly state only devices capable of becoming a coordinator should enter coordinator arbitration. Updated 11.13 Dataflow Cycles to 1) stop sending Node Discovery to address 0x01 if the coordinator happens to be a priority subordinate, 2) stop sending Address Confirmation and Token Offer Broadcast messages if no device in the node list could respond, and 3) explicitly state when the R2R to the virtual internal subordinate occurs. Updated 11.11 AutoNet Server Procedures to identify the AutoNet procedure is aborted when no address space is available. Updated 6.3 Send Methods to indicate coordinator fills in send parameter 2 with a subordinate's index of same node type.

Contributors

The following is a list of ClimateTalk Alliance member companies that were actively engaged in the development of this standard:

A.O. Smith Water Products Company

ecobee inc.

EDC

Emerson Electric, Co.

EWC Controls, Inc.

Microchip Technologies, Inc.

Nogginhaus, LLC.

Research Products Corp.

Rheem Manufacturing Company

Zonefirst

Table of Contents

List of Figures

List of Tables

1.0 Overview

1.1 ClimateTalk Model

ClimateTalk is an open standard that defines a set of messages and commands to enable interoperability, enhanced user interface, and machine to machine control independent of the physical layer connecting the devices.

The messages and commands defined by ClimateTalk Information Model (CIM) are the presentation and application layers as defined by the OSI Model[1]. ClimateTalk Applications are fully defined at Layer 7 of the OSI model by a combination of a Device Specific Application Profile, the Generic Application Specification and the Command Reference.

ClimateTalk messages can be carried over any physical medium following the OSI model. The ClimateTalk Presentation Layer defines how messages are executed over the various physical mediums in use.

CT-485 and CT-LWP are wired serial physical and network layers designed to support the formation of ClimateTalk networks and transport ClimateTalk messages, but other OSI based protocols – including wireless transports - can be used as well.

1.2 Scope

CT-485 is a Physical, Data Link, and Networking set of specifications that define one of the physical media over which ClimateTalk messages are sent. CT-485 is a variant of EIA/TIA-485[2] standards with provisions against incorrect wiring and grounding requirements that meet the needs of residential systems.

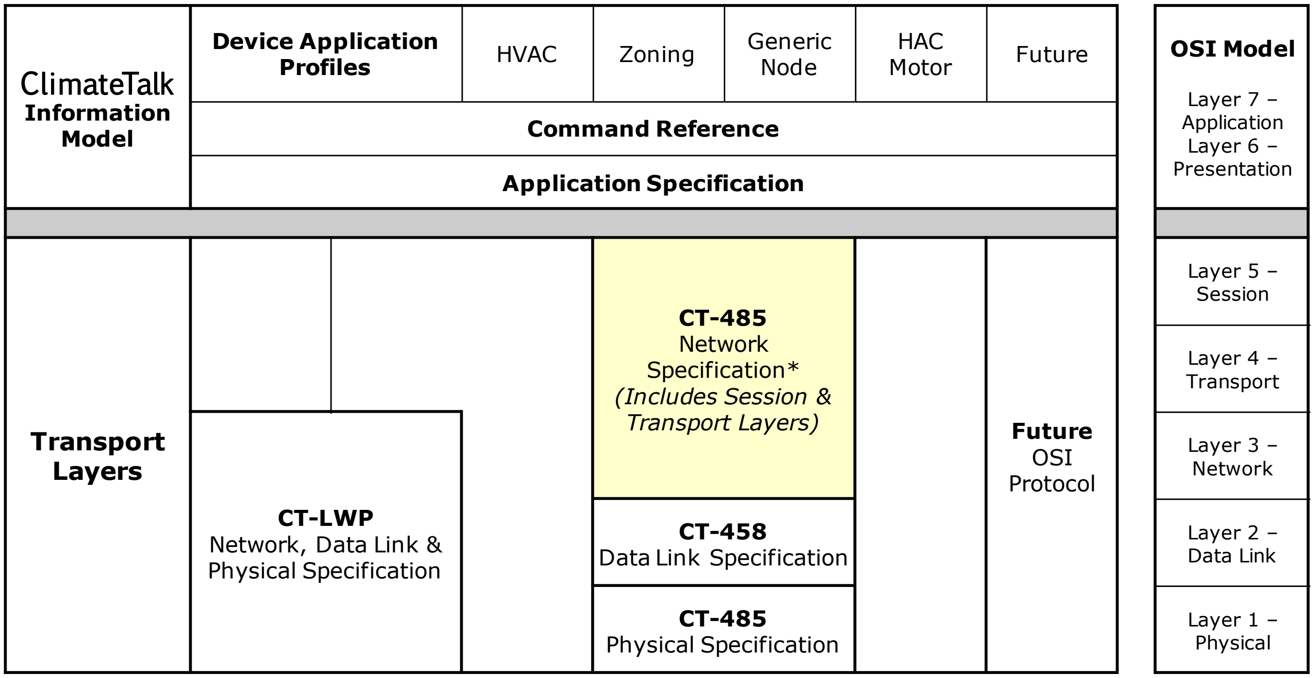
This document defines the Network Layer requirements for CT-485, which corresponds to OSI Layer 3. The Network Layer in CT-485 includes some of the functions performed by the OSI Transport and Session Layers.

Included in the Network Specification is the higher-level network functionality including Routing, Addressing, and Network Timing as well as the requirements for a CT-485 device to operate within a defined network structure.

[1] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20269

[2] http://standardsdocuments.tiaonline.org/tia-tsb-89-a.htm

Figure 1 - OSI Layers for a CT-485 Implementation


ClimateTalk Information Model
Device Application Profiles
HVAC
Zoning
Generic Node
HAC Motor
Future
Command Reference
Application Specification
Transport Layers
CT-LWP Network, Data Link & Physical Specification
CT-485 Network Specification* (Includes Session & Transport Layers)
CT-458 Data Link Specification
CT-485 Physical Specification
Future OSI Protocol
OSI Model
Layer 7 – Application
Layer 6 – Presentation
Layer 5 – Session
Layer 4 – Transport
Layer 3 – Network
Layer 2 – Data Link
Layer 1 – Physical

**This Document*

The ClimateTalk Open Standards package shown in Figure 1 - OSI Layers for a CT-485 Implementation prescribes the mandatory requirements to ensure proper network formation of interoperable devices. Each device must comply with the mandatory requirements defined in this document as well as all other ClimateTalk standards applicable to the device functionality.

Membership in the ClimateTalk Alliance as well as successful completion of mandatory conformance testing is required for listing a product as a ClimateTalk Certified Device.

1.3 CT-485 Version 2.0 Enhancements

To support these additional features, there are some architectural changes in Addressing and Routing. CT-485 Version 2.0 adds the following functionality to the CT-485 Version 1.0 specifications:

- Support for more nodes
    - Required by Zoning and Water Heater Support
    - Up to sixty-two Subordinates over two subnets
    - Recommended maximum number of active nodes increases from eight nodes in V1.0 to forty nodes in V2.0

  - One Coordinator address that exists on both subnets by the same entity
  - New support for multiple devices of the same Node Type
  - All CT-485 V1.0 Subordinates are placed on Subnet 2 and all later Subordinates are placed on Subnet 3
- Support for routing to and source identification of multiple nodes of the same type
  - All three Send Methods (0, 1, and 2) from V1.0 still are supported
  - V2.0 now uses Send Parameter 2 to identify the specific sending Node Index when multiple nodes of the same type are present
  - Send Parameter 2 in V1.0 devices remains set to 0 for backwards compatibility with V1.0
  - New Send Method 3
    - Sends to specified Socket via Coordinator
- More sophisticated Network Coordinator Arbitration
  - ClimateTalk continues to require a single Coordinator to control all network traffic
  - Multiple devices on the network can support the Coordinator function and may have a mixture of V1.0 and V2.0+ devices
  - The highest version device must become the Coordinator
  - CT-485 V2.0 incorporates a new mechanism called the "Coordinator Arbitration Version Announcement" to determine and select the highest version Coordinator
- Token Offer Broadcast (TOB)
  - New for V2.0 devices to improve scalability
  - Separation between Keep Alive Messages and the Subordinate Request to Receive mechanism
  - Coordinator sends the TOB Request and the Subordinate with highest need to send responds first
  - Address Confirmation Broadcast, which is new for V2.0, is sent to all V2.0+ nodes every 120 seconds to confirm node presence
- Bandwidth-node count independence, which means that the earlier version of CT-485 had a system of providing R2R to every node in a round robin fashion. This meant the transmission opportunities a node would receive went down depending on how many nodes were present. By moving to a Token Offer Broadcast system, we change this to give nodes that actually have something to send an opportunity to contend for a transmission opportunity in a fair manner by use of a slot delay

2.0 Normative References

A good understanding of the most recent version of the following documents is required to apply the contents of this specification correctly.

ClimateTalk Generic Application Specification

ClimateTalk Command Reference

ClimateTalk CT-485 Application Protocol Interface

ClimateTalk CT-485 Networking Specification

ClimateTalk CT-485 Data Link Specification

ClimateTalk CT-485 Physical Specification

3.0 Terminology

3.1 Definitions

Coordinator An internal system within a device that connects a Coordinator's Controller Application to a CT-485 Network

Coordinator's Controller Application A general network service application that provides the essential network dataflow structure and a select set of network functionality to all the other network devices

Full Functionality Devices Devices whose functionality is inclusive of all functional requirements within the CT-485 Networking Specification

Message Header Provides information about the packet embedded within a CT-485 Message. Information regarding the type of packet, who sent the packet, who should receive the packet and how the packet should be sent, are all contained within this segment of the message.

Priority Device The Priority Device for a given Control Command Code is determined by the Network Coordinator and is determined by looking at prioritized list of Node Types responsible for processing each valid Control Command Code

Priority Subordinate The Priority Subordinate is located at address 0x01 and receives more opportunities to send than any other Subordinate

Reduced Functionality Devices A device whose functionality has been reduced by not supporting specific functional requirements of a Full Functional Device or FFD

Send Method Part of the Message Header used to route Messages to the intended recipient by way of a Network Coordinator

Subordinate An internal system within a device that provides a communications path between Subordinate User Applications and the ClimateTalk Network Coordinator

Subordinate User Applications The end user applications that communicate within a CT-485 system

3.2 Acronyms

AC	Air Conditioner
ACK	Acknowledge
ADEE	Application Microcontroller EEPROM Shared Data
AH	Air Handler
EEPROM	Electrically Erasable Programmable Read-Only Memory
FFD	Full Functionality Devices
HP	Heat Pump
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IFC	Integrated Furnace Control
MAC	Media Access Control
MC	Memory Card or Memory Stick Shared Data
MTR	Motor Scratch Pad
NAK	Negative Acknowledge
ND	Network Shared Data
XOVER	Crossover device (Formerly known as On-Board Bus Interface /OBBI)
OEM	Original Equipment Manufacturer
R2R	Request to Receive
RFD	Reduced Functionality Device
TBD	To Be Determined

3.3 Word Usage

The conventions used in this document are modelled after the definitions of the *2009 IEEE Standards Style Manual*. The *IEEE Standards Style Manual* can be downloaded from https://development.standards.ieee.org/myproject/Public/mytools/draft/styleman.pdf.

can Equivalent to *is able to* or *is capable of*.

may Equivalent to *is permitted to* or *is allowed to*. The use of *may* means that something is optional and does not imply a requirement.

must Used to describe situations where no other course of action is possible.

shall Equivalent to *is required to*. Use of the word *shall* means that the specification shall be implemented exactly as described in order to ensure correct operation and interoperability with other devices.

should Equivalent to *is recommended that*. This is used in situations where there are several possible options, but one option is preferable to the others

4.0 Network Device Classifications

Configured devices are classified as a Network Subordinate or as a Network Coordinator. The CT-485 network architecture requires that one network device be capable of acting as a Master Network Controller device, called a Network Coordinator. All other devices are slave devices called Subordinates. The Network Coordinator has the added ability that it also can communicate with a virtual device known as a Virtual Subordinate.

CT-485 V2.0 support:

- One Priority Subordinate at address 0x01 on either Subnet 2 or Subnet 3
- Up to fourteen CT1.0 Subordinates on Subnet 2 (Addresses 0x01 – 0x0E)*
- Up to forty-eight CT2.0 Subordinates on Subnet 3 (Addresses 0x01, 0x10 – 0x3E)*
- Addresses in use on Subnet 2 and 3 shall not overlap
- One Coordinator Address (Address 0xFF) that exists on both subnets by the same entity
- One Virtual Subordinate (within the Coordinator)
- A recommended maximum of forty active Subordinates (including the Virtual Internal Subordinate)
- All CT-485 V1.0 Subordinates shall be placed on Subnet 2 and all later Subordinates shall be placed on Subnet 3

* The number of unique Subordinates (as indicated by different node type) is limited to fifteen when a CT1.0 Subordinate is on the network. Reference Section 8.1.2.4.

4.1 Network Coordinator

A Coordinator is an internal system within a device that connects a Coordinator's Controller Application to a CT-485 Network. The Coordinator's Controller Application is a general network service application that provides the essential network dataflow structure and a select set of network functionality to all the other network devices.

In CT-485, there is only one active Coordinator in a system at any given time with a network address of 0xFF. Since multiple devices can be capable of becoming the Coordinator and there are multiple versions of CT-485, it is imperative that the Coordinator Arbitration system ensures that the device with the latest and greatest version of ClimateTalk that is capable of becoming a Coordinator actually becomes the Coordinator for the network. This process is defined in Section 11.4.

4.2 Network Subordinate

A Subordinate is an internal system within a device that provides a communications path between a Subordinate User Application and the CT-485 Network Coordinator. This communication path can use a physical network communications media or can be an internal software communication path.

Subordinate User Applications are the end user applications that communicate within a ClimateTalk system. Since every device on the network shall implement a Subordinate User Application, every device in CT-485 must be capable of acting as a Network Subordinate.

In CT-485, there must be at least one Subordinate on the network to facilitate active communications, provided there is an acting Coordinator in the system. With only one Subordinate, this device's Subordinate User Application is allowed to communicate with the active Coordinator's Controller Application. Since Subordinate User Applications generally talk to other Subordinate User Applications, a second Subordinate typically is needed for a truly functional network.

A Subordinate device is required to validate communications with its Internal Subordinate User Application before becoming active on the network. A device that cannot communicate internally with the Subordinate User Application is likely to be in a failure state or a reset state, or at the very least, is going to cause an immediate failure state or reset state once network dataflow transactions begin.

4.2.1 Subordinate Address Assignment

CT-485 allows one Priority Subordinate to be at logical address 0x01 on Subnet 2 or Subnet 3. The only Node Types allowed to be a Priority Subordinate device are Node Type 1 (Thermostat) and a Node Type 21 (Zone Controller).

In CT-485, address assignment is handled in two separate pools of addresses on Subnet 2 and Subnet 3. The addressing order for Subnet 2 goes from 0x01 to 0x0E with the caveat that address 0x01 on Subnet 2 is assigned only if the node being addressed is appropriate for the Priority Address i.e. a Thermostat or Zone Controller. Subnet 3 addressing order starts with address 0x01, again applicable only if node-appropriate, but then goes on to addresses 0x10 through 0x3E.

An AutoNet Server handles the Address Assignment in CT-485, which is co-located with the Network Coordinator. The AutoNet Client co-located with each Subordinate is what allows the AutoNet mechanism to allocate addresses dynamically to Subordinates. Refer to Sections 11.11 and 11.12 for details on how this is accomplished.

4.3 Virtual Internal Subordinate

Virtual Internal Subordinate devices are part of a conceptual software model that is implemented in CT-485 to provide a universal interface to end user applications. If a device is the active Coordinator in the system, a Virtual Subordinate device provides a software link between the Subordinate User Application and the local Coordinator's Controller Application. This software link merely emulates the transactions that normally would take place over the communication media.

A Virtual Subordinate is tied to the Coordinator and will be identified to all network Subordinates via a Node List Notification just like any other active Subordinate device. An active Virtual Subordinate is identified by the first entry in any Network Node List Notifications sent by the Coordinator. Virtual Subordinates maintain a logical address of 0xF1, which is available on both subnets and is utilized internally to address Messages delivered via the internal software link instead of external communications.

If an FFD that knows it has an Internal Subordinate but cannot establish a link to it, or verify its presence, may still choose to attempt Coordinator Arbitration or choose to not attempt to be Coordinator until communication with its Internal Subordinate application is established. However, it is preferable that an FFD attempt Coordinator Arbitration, even if it cannot establish communication with the Internal Subordinate, to improve the fault tolerance of the network. If an FFD becomes Coordinator without an Internal Subordinate, it shall report a '0' in the Node List for the position of the Internal Subordinate.

4.4 Full Functionality Devices (FFD)

Full Functionality Devices or FFD(s) are devices whose functionality is inclusive of all functional requirements within the CT-485 Networking Specification. Full Functionality Devices must be capable of acting as a Network Subordinate as well as a Network Coordinator if needed. When an FFD is acting as the Network Coordinator, it utilizes the Virtual Subordinate model in order to allow the device to also be an active Network Subordinate.

To ensure that there is at least one Coordinator-Capable Device on the network, it is mandatory that all indoor units (Node Type 2 and Node Type 3) be Full Functionality Devices. In addition, a Zone Controller, if present, is also required to be a FFD on the network.

4.5 Reduced Functionality Devices (RFD)

Reduced Functionality Devices or RFD(s) are devices that do not support Network Coordinator or AutoNet Server functions. All devices require the ability to act as a Subordinate to support an end user application; therefore, all RFDs shall be capable of acting as Subordinate devices.

4.5.1 Effects of Reduced Functionality Devices

Although limited, there are scenarios in which RFDs cause harm to a system.

4.5.1.1 Coordinator Loss Effect

Since only one Coordinator-capable device is required to run a CT-485 network, it theoretically is possible to have a network full of RFD Subordinate devices and only one Coordinator-capable device. Subordinate-only devices place a strain on the network. The strain on the network decreases as more Coordinator-capable devices are added and increases as more Reduced Functionality Devices are added.

All ability to move data on the network stops when there is no Coordinator available. This leaves all RFD Subordinate devices "cut off" in unknown states until a new Coordinator initiates contact.

4.5.1.2 Strain Situation Scenario 1 – Best Case

A situation in which there is only one RFD Subordinate-only device comes at the least cost to the network because it has almost no downside. In general terms, the network would have to fail down to just two nodes capable of becoming the Coordinator (an RFD and a FFD node) before the possibility for a central failure point occurs. This scenario may not have any noticeable effects on cases in which three or more nodes are necessary to have a fully functional system since the functional system breaks down before the network would.

4.5.1.3 Strain Situation Scenario 2 – Worst Case

A situation in which only one Coordinator capable device is present is the most taxing to the stability of the network. In this scenario, there is a central failure point at the Coordinator device. If the Coordinator-capable device fails to negotiate or fails to remain as the Network Coordinator, then the network fails.

4.5.2 Connectivity Loss Effect

All Subordinate devices on a CT-485 Network must be able to communicate with the Network Coordinator in order to function properly. In most situations, this is not an issue since every device is capable of direct communication with the Network Coordinator. In normal scenarios, if a CT-485 Network Coordinator is not able to maintain its link to all Subordinates, the nature of the system corrects this by moving the Coordinator around to a device capable of maintaining a link to all network Subordinates.

In a network that contains RFDs, it could become impossible to move the Coordinator to an appropriate device that is capable of communications with all Subordinate devices on the network.

This Connectivity Loss Effect only is relevant to networks where a pre-existing condition exists in which network devices do not have a direct communication path between each other. In other words, there must already be a communication problem in the network before the network can have a Connectivity Loss Effect.

5.0 Frame Format

The CT-485 Message Frame is classified into four basic segments:

a. Message Header
b. Packet Header
c. Packet Payload
d. Message Footer

Further, the Message Frame is classified into eleven individual elements.

Table 1 – CT-485 Message Structure Elements

Element	Segment	Size (in bytes)	Starting Offset in Hex
Destination Address	Message Header	1	0x00
Source Address		1	0x01
Subnet		1	0x02
Send Method		1	0x03
Send Parameters		2	0x04
Source Node Type		1	0x06
Message Type	Packet Header	1	0x07
Packet Number		1	0x08
Packet Length		1	0x09
Packet Payload	Packet Payload	0-240	0x0A
Message Checksum	Message Footer	2	0x0A + Packet Length (directly after packet payload)

5.1 Message Header Elements

Message Header Elements provide information about the packet embedded within a CT-485 Message. Information on what type of packet, who sent the packet, who should receive the packet and how the packet is sent are all contained within this segment of the message. The Network Layer is responsible for the contents of the message header (shown in blue in Table 1 – CT-485 Message Structure Elements).

CT-485 systems process the Message Header Elements to determine which packets to filter from further processing. By only allowing messages that are addressed appropriately for a particular application to be processed by that application, the burden on the application to validate packets is greatly reduced. This provides a layered structure between the initial message processor and the user's application.

5.1.1 Destination Address

The Sending Node fills the Destination Address byte within the Packet Header with the address of the intended recipient. Each Receiving Node checks this Destination Address. A Receiving Node only processes packets whose Destination Address matches its local address and whose destination subnet matches its subnet. A Destination Address of zero (0) indicates that a packet is a broadcast packet. Every node on the destination subnet should process broadcast packets.

5.1.2 Source Address

The Source Address field within the Packet Header is filled with the node's own address by the Sending Node. The Source Address is used by the Receiving Node to know where to direct a response.

5.1.3 Destination Subnet

The Subnet field within the Packet Header is filled with the Subnet of the intended recipient by the Sending Node. Each Receiving Node checks this Subnet. A Receiving Node only processes packets whose Subnet matches its local subnet.

5.1.4 Send Method and Send Parameters

The Send Method is how messages are routed between Subordinate devices. The parameters used depend on the Method Type. For information on the contents of these fields, see Section 6.3 Send methods.

5.1.5 Source Node Type

The Source Node Type within the Packet Header is filled by the original Sending Node with the Node Type of the Sending Node. The Node Type is unchanged by the Coordinator, even if the packet is being routed. The Node Type can be used for secondary validation that a node at a certain address is of the Node Type that was expected to be at that address. The Coordinator, as one check to maintain the current Session, does this validation. This is important if nodes have been addressed by the Coordinator, then go off-line for a period, and then come back on to the network after the Coordinator has addressed a new node with the same address.

5.2 Packet Header Elements

Packet Header Elements provide additional information about the packet embedded within a CT-485 Message Frame. The Application Layer is responsible for the contents of the Packet Header. See the Generic *Application Specification* for more information.

5.3 Packet Payload Element

The Packet Payload Element provides the application a space to read data (on reception) or write data (when preparing a transmission). This information is the actual data sent on a CT-485 network by its users. For more information, see the *Generic Application Specification*.

5.4 Message Footer Element

The Message Footer provides a checksum of the entire Message Frame that provides a significant level of data integrity. The Data Link Layer is responsible for the Message Footer. See the *CT-485 Data Link Specification* for more information.

6.0 Routing

6.1 Node Address

The table below lists the complete CT-485 address range.

Table 2 – CT-485 Address Ranges

Address		Size	Reserve Range Purpose
Decimal	**Hexadecimal**	**Bytes**	**Purpose**
0	0x00	1	Broadcast
1 – 14	0x01 – 0x0E	14	CT1.0 Subnet 2 Subordinate Addresses (7 Active Node Maximum)
15	0x0F	1	Reserved for Future Use (Overhead Validation)
16 - 62	0x10 – 0x3E	47	>CT1.0 Subnet 3 Subordinate Addresses
63	0x3F	1	Reserved for Future Use (Overhead Validation)
64 - 84	0x40 – 0x54	21	Reserved for Future Use (Internet Access)
85 - 90	0x55 – 0x5A	6	Restricted Address Range (Diagnostic Use)
91 – 191	0x5B – 0xBF	101	Reserved for Future Use (Wireless Address Range)
192	0xC0	1	Restricted Address (Network Analysis)
193 – 207	0xC1 – 0xCF	15	Reserved for Future Use (System Authentication)
208 – 239	0xD0 – 0xEF	32	Reserved for Future Use (Bridge/Routing Address Range)
240 – 253	0xF0 – 0xFD	14	Reserved for Future Use (RFD Devices)
254	0xFE	1	Coordinator Arbitration address
255	0xFF	1	Network Coordinator Address

6.2 Subnets

There are two subnets used for CT-485. Subnet 2 is used for all CT-485 V1.0 Subordinates and Subnet 3 is used for all later-revision devices. To send a Message to all subnets, a value of zero (0) is used. Other subnet addresses are reserved for future use.

6.3 Send Methods

All commands sent in a CT-485 Network use the Destination Address and Subnet to determine which device will receive a given message. Subordinate devices are only allowed to send traffic to the active Network Coordinator. To send to another Subordinate, a message must be routed through the Coordinator. Multiple methods exist for forwarding commands through the Coordinator.

The Coordinator routes response Messages back to the original requesting device. The Send Method and Send Parameters should be copied to the Response Message from the original Request Message.

The Send Method is in the Message Frame Header and is filled (or cleared) by the sender of the Message Frame. Several supported Send Methods are associated with specific routing mechanisms.

If no device is found that matches the specified Send Parameters for Routing in any of the Send Methods, then the message will not be routed. The Coordinator discards it.

6.3.1 Send Method 0: Non-Routed Messages

If the Send Method is equal to zero (0), the Coordinator does not forward this packet.

a. Send Method	=	0
b. Send Parameter	=	0
c. Send Parameter 2	=	Coordinator fills Send Parameter 2 with the requesting device's 'Index of Source Node among nodes of its own type' when sending a Send Method 0 message to a Subordinate. A Subordinate shall populate Send Parameter 2 field with zero (0) for all Messages originating from it.

Send Method 0 transactions are referred to as Non-Routed Sends. Non-Routed Sends are addressed directly to the Destination Device and do not require further routing to reach their destination. To indicate this, the Sender must fill the Send Method and Send Parameter 1 with zero (0) to indicate that the message is not to be forwarded by the device that receives it. The response to a Non-Routed Send shall populate the Send Method and Send Parameters with the same values received during the request.

In all messages with the Send Method equal to zero (0), the Coordinator shall send the Subordinate its index within devices of the same type. For example, if there are four nodes of the same type, the first node in the Node List of that type will always see a value of zero (0) in the Send Parameter 2 field for all of the Send Method 0 packets received from the Network Coordinator. The third node will see Send Parameter 2 as two (2). If only one instance of a Node Type exists on a network, it always will see a zero (0) for Send Parameter 2, which maintains backward compatibility.

When the coordinator sends a message to a Subordinate using Send Method 0, the coordinator shall populate Send Parameter 2 with the Subordinate's index within nodes of same type. The Subordinate application is informed by the Coordinator of its Index within nodes of the same type when receiving an addressed network node list message. The Subordinate shall populate Send Parameter 2 field with zero (0) for all Messages originating from it.

6.3.2 Send Method 1: Routing by Priority Control Command Device

The Subordinate initiating the transaction addresses all messages with Send Method 1 to the Coordinator. The Coordinator, as explained in this section, handles routing.

If the Send Method within a message is one (1), the packet is forwarded to the Priority Device (using a Priority Table) based on a code that is held in Send Parameter 1. Example codes for Send Parameter 1 are Priority Heat Device, Cool Device, etc.

a.	Send Method	=	1
b.	Send Parameter	=	Control Command Code
c.	Send Parameter 2	=	Device initiating request fills Send Parameter 2 with 0. Coordinator fills Send Parameter 2 with the requesting device's 'Index of Source Node among nodes of its own type' when this message is routed to the destination device. Coordinator fills Send Parameter 2 with 0 when routing the response back to the requesting device.

The Priority Device for a given Control Command Code is determined by the Coordinator and is determined by looking at a prioritized list of Node Types responsible for processing each valid Control Command Code. The device with the highest Node Type in the prioritized list for a particular Control Command Code will be the Priority Device for that Control Command Code. If no suitable device is found, then the command will not be routed. Consult the *Command Reference* for relevant Control Command Codes.

Table 3 below shows the supported Control Command values that may be used as Send Parameter 1 in a Send Method 1 message and the priority of each Node Type in relation to the Control Command Code.

NOTE 1: Send Method 1 does not support routing to a specific node within nodes of the same type. If a Send Method 1 Message resolves to a Node Type, for which multiple instances of that Node Type may exist on the network, the Message is routed to the first node of that type within the Node List.

Table 3 – Priority Device Routing

Send Method 1 Priority List							
Control Command Description	Command Code		Device Priority				
	Decimal	Hex	1	2	3	4	5
Heat	100	0x64	ZC	HP	IFC	XOVER	AH
Cool	101	0x65	ZC	HP	AC	XOVER	IFC
Fan	102	0x66	ZC	AH	IFC	XOVER	-
Emergency	103	0x67	ZC	AH	IFC	XOVER	-
Defrost	104	0x68	ZC	AH	IFC	XOVER	-
Aux Heat	105	0x69	ZC	AH	IFC	XOVER	-

6.3.3 Send Method 2: Routing by Priority Node Type

All messages with Send Method 2 are addressed to the Coordinator by the Subordinate initiating the transaction. This section explains how Routing is handled by the Coordinator.

If the Send Method is 2, then the packet will be forwarded by the Coordinator to the Priority Device having the same Node Type as indicated by the value held in the first byte of the Send Parameters. Examples are the Priority Heat Pump, Air Handler, etc.

a. Send Method = 2
b. Send Parameter = Targeted Node Type
c. Send Parameter 2 = Device initiating request fills Send Parameter 2 with 0.

Coordinator fills Send Parameter 2 with the requesting device's 'Index of Source Node among nodes of its own type' when this message is routed to the destination device.

Coordinator fills Send Parameter 2 with 0 when routing the response back to the requesting device.

The Priority Device for a given Node Type is simply the first instance of that Node Type that the Coordinator finds when searching through its list of network devices.

NOTE 2: Send Method 2 does not support routing to a specific node within nodes of the same type. If a Send Method 2 Message uses a Send Parameter of 1, for which multiple instances of that Node Type may exist on the network, the Message is routed to the first node of that type within the Node List.

6.3.4 Send Method 3: Routing by Socket

The Subordinate initiating the transaction addresses all Messages with Send Method 3 to the Coordinator. This section explains how the Coordinator handles Routing.

If the Send Method is 3, then the packet is forwarded by the Coordinator to the device that is at the position in the Node List, indicated by the Send Parameter 1 value. This value is called the Socket and the Routing Method shall be called Routing by Socket.

a.	Send Method	=	3
b.	Send Parameter	=	Socket (Node List position) of Destination Node
c.	Send Parameter 2	=	Device initiating request fills Send Parameter 2 with 0. Coordinator fills Send Parameter 2 with the requesting device’s ‘Index of Source Node among nodes of its own type’ when this message is routed to the destination device. Coordinator fills Send Parameter 2 with 0 when routing the response back to the requesting device.

Note: The Node List position is with respect to the Node List sent to CT2.0 devices, not the “condensed” Node List sent to CT1.0 devices. Reference paragraphs 8.1.2.3 and 8.1.2.4.

6.4 Send Parameters

The Send Parameter fields are modifier values for the Send Method and are filled (or cleared) by the Sending Device. The use of Send Parameters to support Send Methods is detailed in the previous sections.

7.0 Network Design

7.1 MAC Address

Each CT-485 device is assigned a MAC Address that provides a unique identifier for each device. The sender's MAC Address is embedded into the payload of poll messages and poll acknowledgements. See the *CT-485 Data Link Specification* for the format of the MAC Address.

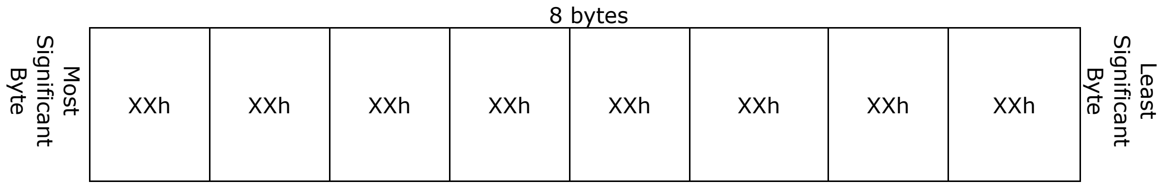
7.2 Session ID

Each CT-485 device must be capable of generating a pseudorandom, unique 8-byte value to use as a Session ID for each communicating application. An application can be a Subordinate User Application, a Coordinator Application or both a Subordinate and a Coordinator Application.

The Session ID may be changed at any time, declaring a new operational state. The Coordinator uses the Session ID along with the MAC Address to maintain the operational state of all the nodes in its subnet.

Session IDs of all zeros (0) are not allowed. At least one bit must be set in the 8-byte field.

Figure 2 – Session ID Size Diagram



7.3 Network Initialization

The Coordinator checks periodically for new nodes on the network. When the node has completed the steps necessary for getting on the network, the Coordinator shall transmit a Set Network Node List providing an application a list of active nodes on the network.

If at any time during operation the device receives a Network Node List, it shall assume that something has changed on the network. The device shall return to the same state as when it was first activated on the network and suspend all current operations. Devices shall reissue any commands as required based on the new network status.

7.4 Addressed and Broadcast Messages

CT-485 messages are classified as either Addressed or Broadcast. Any Message shall be either Addressed or Broadcast. When it is a Broadcast Message, any Message may be broadcast to all addresses within a subnet by using an Address Value of 0 and a specific subnet value, or it may be addressed to all nodes in all subnets by using a 0 for both Address and Subnet.

7.4.1 Addressed Message Rules

All Addressed Messages received with a valid checksum shall be acknowledged immediately with an ACK. All addressed Request Messages, which have been ACK'd, shall be responded to with the corresponding Response Message at the next transmission opportunity.

7.4.2 Broadcast Message Rules

Only the Coordinator is allowed to send Broadcast messages. No acknowledgement is sent by a Subordinate receiving a Broadcast message. If any of the Receiving Nodes matches the requirements to send a response to the received Broadcast Message, it shall send the response after a Slot Delay (explained in Section 11.1). Further, to avoid collisions, any node that sees the first node's response begin on the bus will back off and abort its own response.

7.5 Network Dataflow Design

In CT-485, only one device is allowed to transmit data at a time. The Coordinator is responsible for controlling the flow of traffic by polling each device in turn and giving it an opportunity to transmit data to another device. The polling message is called a "Request to Receive" or R2R for short. Each Subordinate shall wait to receive a properly formatted R2R message before sending any messages to the network.

Subordinates are required to respond to all Addresses Messages, including R2Rs, which the Coordinator sends to them. This allows the Coordinator to move on to the next transaction more quickly than having to wait for a timeout to occur.

If a Subordinate receives an R2R and has no data to send, it shall respond with an R2R acknowledgement (ACK) message. For the appropriate Message structure, see the ClimateTalk *CT-485 API Reference*.

If a Coordinator has a Request or Response Message for a device, it may send the request at any time that it is not actively involved in a conversation with another device. Coordinator Request and Response Messages are not preceded by an R2R message. The Receiving Device shall acknowledge Requests and Responses. The Coordinator shall drop devices that fail to respond to three consecutive Request or Response Messages. For more information on acknowledgements, see the following sections.

7.5.1 Request to Receive (R2R) Messages

When Subordinate devices are brought onto the network, they learn the MAC Address of the Coordinator and its Session ID. Subordinates use this information to ensure that messages they receive are from the correct Coordinator and for the current session.

Each R2R message contains the MAC Address of the Coordinator and the Coordinator's current Session ID. The Subordinate compares the MAC Address and Session ID of the R2R with the MAC Address and Session ID received when the device was configured on the network. If the Subordinate receives an R2R with a MAC Address or Session ID that differs from the correct MAC Address and Session ID, the Subordinate will not send any new messages and will wait to be renegotiated by a new Coordinator.

7.5.2 Request to Receive - Acknowledgements (R2R-ACKs)

R2R Acknowledgements (ACKs) contain the MAC Address and Session ID of the Subordinate. The Coordinator can use this to track unexpected changes to Subordinate devices in the network. If a Coordinator receives an R2R ACK from a different Subordinate than expected, then that Subordinate is re-negotiated by the Coordinator.

7.5.3 Request and Response Acknowledgements (ACK)

On the CT-485 network, the Receiving Node shall acknowledge every Non-Dataflow Message sent between a Transmitting Node on the physical medium directly to a Receiving Node by sending an ACK. The ACK sent for a Non-Dataflow Request or Response shall be similar to the R2R-ACK message, except the Message Type of the ACK shall mirror the Message Type of the original Request or Response being ACK'd. The Coordinator shall send an ACK for every Request or Response Message from a Subordinate and every Subordinate shall send an ACK for a Request or Response seen from the Coordinator. The acknowledgements shall be on a single-hop basis.

7.5.4 Token Offer Broadcast (TOB) Messages

In CT 485 V1.0, R2R messages were issued periodically to all Subordinates to determine if any of them required an opportunity to transmit data. Subordinates also checked to see if they received a directly Addressed Message within the last 120 seconds and dropped off of the network if they had not. Since a Coordinator does not have something to transmit all of the time, the cycling R2Rs also are used to ensure that Subordinates see packets directly addressed to them often enough to stay on the network.

Since this does not scale well as the network expands, the Keep-Alive mechanism is separated from the Transmission Opportunity mechanism for all CT-485 V2.0 and above Subordinates. To manage this, all CT-485 V1.0 Subordinates are placed on Subnet 2 and provided R2R Messages for both Transmission Opportunity and Keep Alive Messages. However, all CT 485 V2.0 and above Subordinates are placed on Subnet 3. To ensure Keep Alive, CT-485 V2.0 incorporates a new Broadcast Message known as "Address Confirmation Broadcast."

A new Broadcast Message known as "Token Offer Broadcast" is defined in CT-485 V2.0 for facilitating the Opportunity to Transmit sequence. A Token Offer Broadcast is sent to Subnet 3 periodically. All Subordinates that require a Transmission Opportunity use a Slot Delay-based arbitration mechanism for responding to this Message.

Among all Subordinates that require a Transmission Opportunity, the one that comes up with the smallest Slot Delay wins and sends a Response to the Token Offer. The Coordinator then provides this device with an R2R, which it uses to send any Messages out to the network.

A subordinate that wins Token Offer Broadcast arbitration shall refrain from responding to the next Token Offer Broadcast during the same dataflow cycle. The dataflow cycle is defined in Section 11.13.

7.6 Coordinator – Subordinate Communications

When a Subordinate receives an R2R, one of several scenarios may occur:

a. The Subordinate has no data to send right now
b. The Subordinate has a request to send to another device
c. The Subordinate has a response to send to another device
d. The Subordinate is busy and cannot process the message right now
e. The Message is malformed, bad, or some other error has occurred

A Subordinate may also receive a Request or a Response Message from the Coordinator. Request and Response Messages are not preceded by an R2R message, as the R2R is used solely to ask the Subordinate if it has something to send. The scenarios for these cases are:

a. The Subordinate receives a request from the Coordinator that it can process
b. The Subordinate receives a request from the Coordinator that it cannot process
c. The Subordinate receives a response from the Coordinator that it can process
d. The Subordinate receives a response from the Coordinator that it cannot process

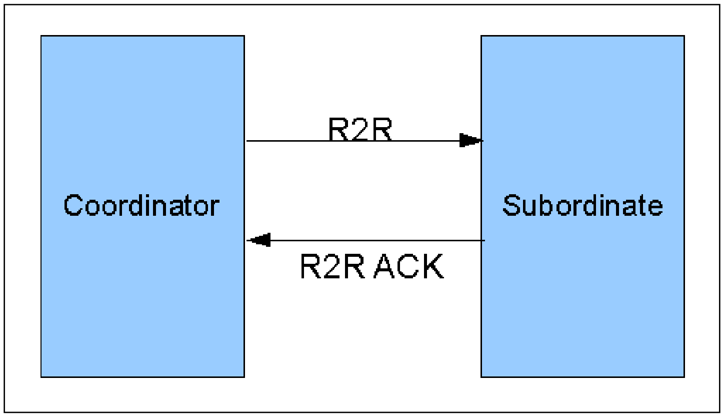
When combined, the R2R and Request/Response Message scenarios form a complete Request/Response Sequence from one Subordinate to another.

Each scenario is illustrated in more detail in the following subsections.

7.6.1 Subordinate Has no Data to Send

When a Subordinate has no messages to send, the Subordinate shall respond with an R2R Acknowledgment (ACK).

Figure 3 – Message Sequence – No Data



If a Subordinate fails to respond to an R2R, the Coordinator may remove it from the network.

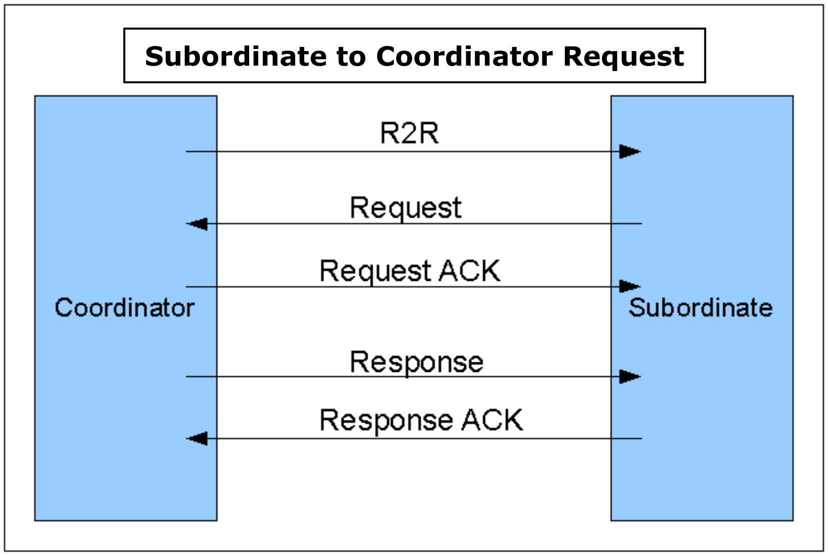
Request to Receive Acknowledgements allow the Coordinator to move on to poll the next device more quickly and allows the Coordinator to validate that the responding Subordinate Device is the expected device and not a new or different one.

7.6.2 Subordinate to Coordinator Request

When a Subordinate has a Request Message to send, the Subordinate shall not send the R2R ACK and shall instead send the request. The Subordinate shall receive a Message Acknowledgment if the message transmission is successful.

The Coordinator shall immediately provide a Response. The Response is not preceded by an R2R.

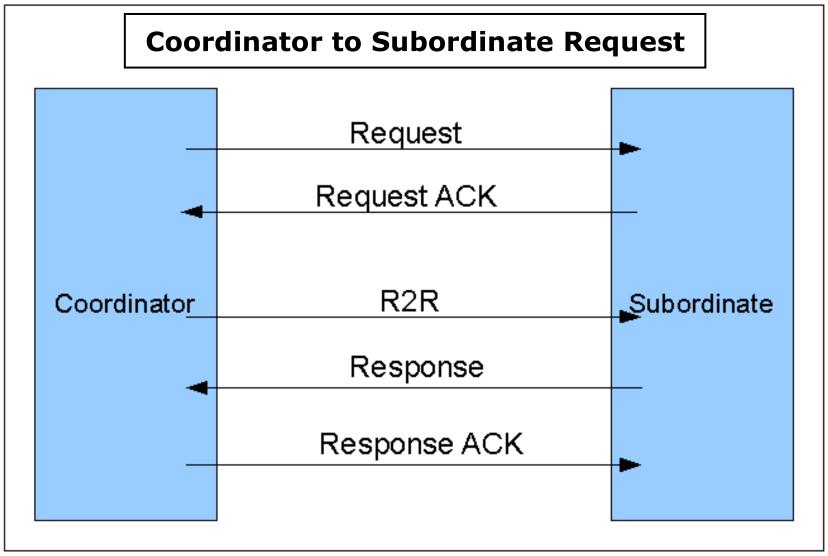
Figure 4 – Subordinate to Coordinator Request



7.6.3 Coordinator to Subordinate Request

If the Coordinator has a Request for the Subordinate, it sends the Request without an R2R preceding it. The Subordinate shall acknowledge the Request. The Coordinator shall immediately send an R2R to the Subordinate to give it the opportunity to respond to the Request. The Coordinator shall not poll other devices between sending the Request and sending the R2R for receiving the Response.

Figure 5 – Coordinator to Subordinate Request



If the Subordinate receives a valid Request that is not understood or for some other reason not processed, then an Error Response shall be generated by inverting the payload of the original Request. This Inverted Payload Response is sent in place of the Response. The original Request still is acknowledged at the Network Layer.

If a Subordinate fails to send a response on the next available opportunity, the network transaction fails. This type of failure is left for the originating Sender to resolve.

7.7 Subordinate to Subordinate Communications

When a Subordinate needs to communicate with another Subordinate, it sends the message through the Coordinator, which routes the message to the correct device.

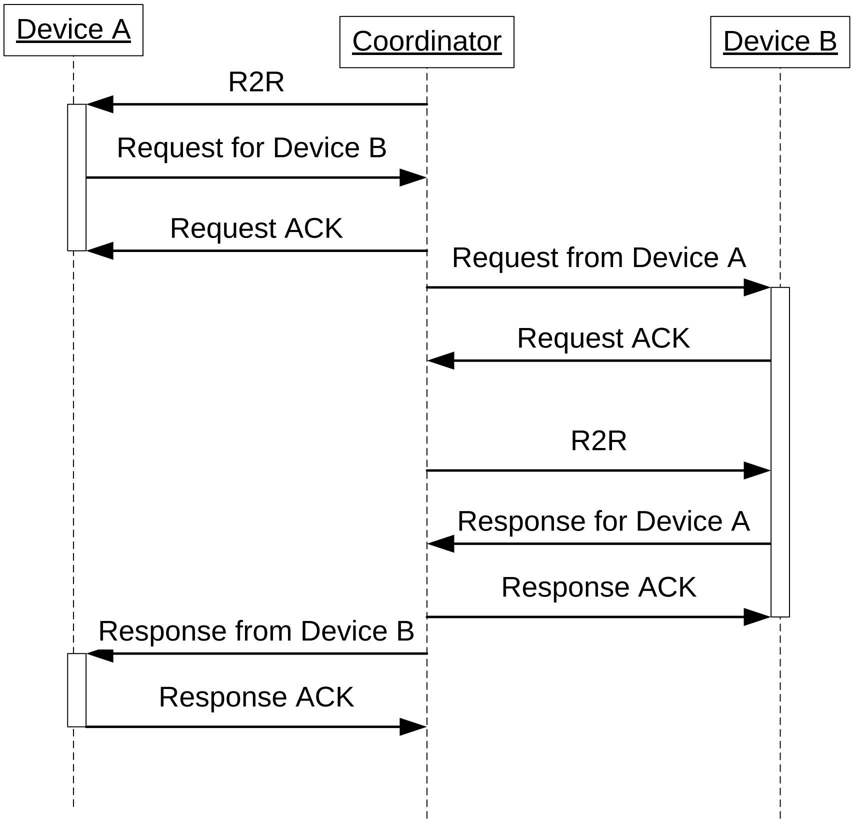
7.7.1 Subordinate to Subordinate Request

A Subordinate must wait to send its Request until the Coordinator polls it. Upon receiving a poll, the Subordinate may send its Request Message. The Coordinator shall respond with an ACK.

The Coordinator then routes the Message to the appropriate device. The device responds with an ACK. The Coordinator immediately issues an R2R to the device to get the Response, which it routes to the original requestor.

The Coordinator shall complete the process of Routing the packet until the requesting Subordinate has received a Response to its Request.

Figure 6 – Subordinate to Subordinate Request



7.8 Timeouts

7.8.1.1 Request to Receive Timeout

The maximum amount of time the Coordinator waits to receive a Response to an R2R before assuming failure is three seconds.

7.8.1.2 Message Acknowledgment Timeout

The maximum amount of time a Subordinate waits to receive an Acknowledgement (ACK) to a Request or Response Message before assuming failure is three seconds.

7.8.1.3 Coordinator Turnaround Time

The maximum amount of time the Coordinator can take between the end of one transaction and the beginning of the next is 500 milliseconds.

7.9 Network Transactions

A network transaction begins with a Request Message and ends with the reception of a Response Message. Coordinators are allowed to initiate network transactions any time there is not already another pending network transaction.

Subordinates can only initiate a network transaction upon receipt of an R2R, provided they do not have a pending Response.

7.9.1 Non-Routed Transactions

A Non-Routed Transaction is a conversation solely between the Coordinator and a Subordinate. Non-Routed Transactions are used by a Coordinator to communicate network information to Subordinates and Subordinates use them to interact with the Coordinator's Network Control Systems.

7.9.2 Routed Transactions

A routed transaction is a conversation between two Subordinates. Since Subordinates are only allowed to send traffic to the Coordinator, the Coordinator acts as a router and routes Messages to the correct destination device.

Unless otherwise noted, Routed Messages require that the Send Method and Send Parameters are copied from the Request Message into the Response Message for the Response to be properly routed back to the originating device.

7.10 Network Sessions

The sender's Session ID is embedded into either the payload of R2R Messages when acting as a Coordinator, or in the R2R Message acknowledgement when acting as a Subordinate. This token represents the logical session for the controlling application. Changes in a Session ID represent changes in an application's session on the network.

Session changes to all devices are communicated to all other devices on the network through the Coordinator. Any time a session change occurs, the Coordinator sends a Network Node List to each device on the network.

Following the logic above keeps all nodes alerted to any session changes within the network and allows devices to respond to these changes.

7.10.1 Coordinator Session Changes

A new Session ID is generated for that application instance each time a Coordinator starts up. This usually means that a device has reset or completely restarted.

Subordinates will track changes to the MAC address and Session ID of the Coordinator. If a Subordinate detects a change in the Coordinator's MAC address or Session ID, the Subordinate generates a new Session ID to acknowledge the change of the Coordinator or change of the Coordinator's session.

7.10.2 Subordinate Session Changes

Each time a Subordinate starts up, a new Session ID is generated for that application instance. This usually means that a device has reset or restarted recently. The controlling application also is capable of forcing a change in the active Session by generating a new Session ID for other reasons.

The Coordinator tracks the MAC address for each Subordinate on the network. If a Coordinator detects a change in the Subordinate's MAC Address, the Coordinator issues new Network Node List Notifications with this device removed from the node list and then proceeds to force this device to lose its address per section 9.1.2.

The Coordinator tracks the Session ID for each Subordinate on the network. If a Coordinator detects a change in a Subordinate's Session ID, the Coordinator issues Network Node List Notifications. The subordinate with the session ID change remains in the Network Node List Notifications. The new node list simply identifies some session has changed.

8.0Network Applications

Network Applications are the internal systems linked to a Coordinator or Subordinate that wish to use the network to send Messages. Network Applications do not send dataflow messages, only normal Request/Response Messages.

Figure 7 – Example Network Hierarchal Model

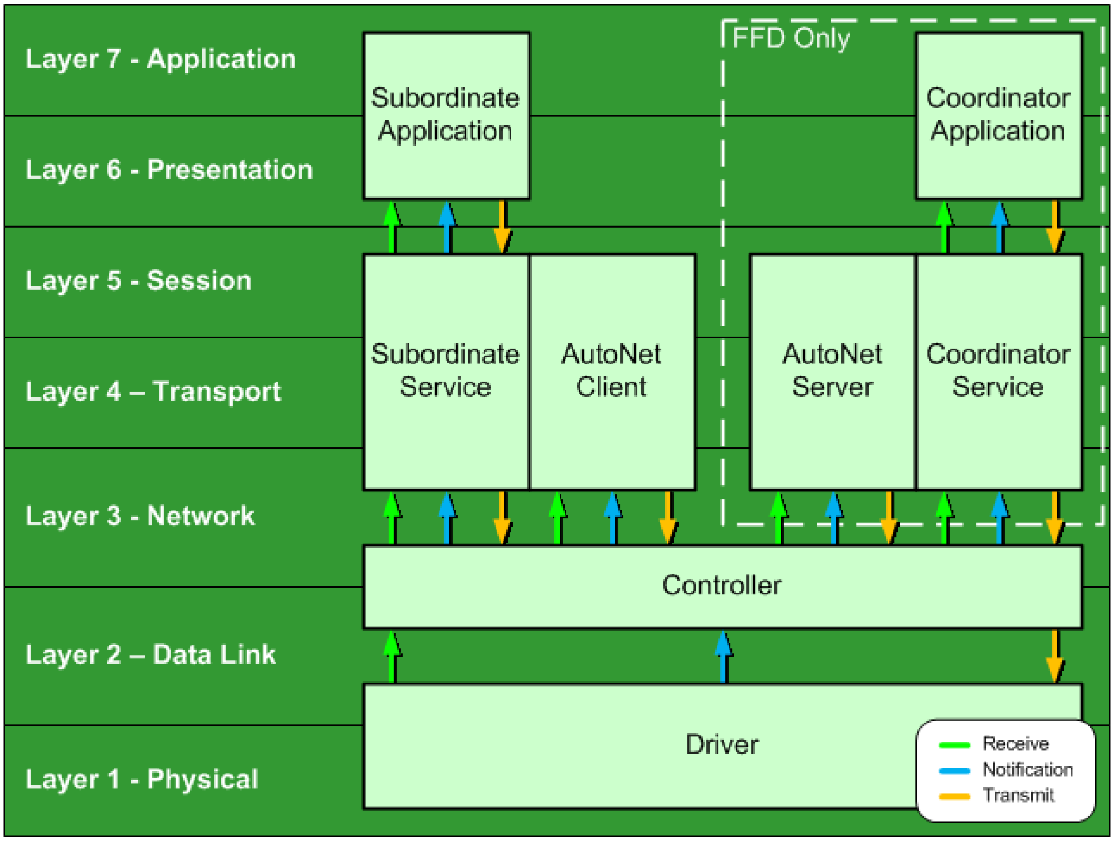


Figure 7 above depicts the general hierarchy of an example network model with respect to the OSI 7 Layer Model. This implementation example contains only four layers that span all seven layers of the OSI Model.

The CT-485 Driver Level interfaces with the actual hardware and resides in the OSI Physical Layer and partially in the Data Link Layer.

Above this, the Controller processes the information from the Driver and either passes it upward to the services or responds with Network Dataflow Messages. The Controller shares the OSI Data Link Layer with the Driver and the OSI Network Layer with the services.

The services are responsible for their reserved range of Messages and maintaining the Network Session. For a Reduced Functionality Device (RFD), there are two services:

a. Subordinate Service

b. AutoNet Client

In addition to these services, a Fully Functional Device (FFD) has a Coordinator Service and an AutoNet Server. The services span the OSI Session and Transport Layers and share the Network Layer with the Controller.

The highest levels of this example are the Network Applications: the Coordinator application and the Subordinate application. These services only send normal Request Messages and Response Messages and contain the Network Taskers. The Network Applications span the OSI Presentation and Application Layers.

8.1 Coordinator Applications

Certain Coordinator Applications are requirements for a functional system. Coordinator Applications are reserved for the purpose of providing network control or additional network functionality.

8.1.1 Coordinator Node Type

The Network Coordinator is always Node Type 165 decimal (0xA5 hexadecimal).

NOTE 1: This Node Type differs from the Internal Virtual Subordinate’s Node Type.

8.1.2 Network Node List Update Notifications

The Coordinator has the responsibility for monitoring all devices on the network and looking for changes to the network configuration. Any time the Coordinator determines that a configuration change has taken place, it stops normal network operations and sends a new Network Node List to all devices on the network. The new Network Node List contains the most up-to-date network configuration.

All devices are responsible for processing and responding to a Network Node List request from the Coordinator. When a device receives a Network Node List, it should follow the rules that are outlined in its device profile for this type of message.

While CT-485 is limited to fewer nodes than the available data packet size, all Subordinates must accept a Network Node List of up to two hundred and forty bytes.

8.1.2.1 Reasons to Issue Network Node List Updates

The following subsections detail the possible reasons for issuing updated Network Node Lists.

8.1.2.1.1 New Device Added to Configuration

When the configuration Tasker validates a newly added device as specified in Section 8.1.2.2, the newly validated device is added to the Network Node List and the new Network Node List update is sent out to all devices on the list, including the newcomer.

8.1.2.1.2 New Session for a Device

When one of the active devices populates an ACK with a different session ID from what it had been using until then, a Network Node List update is sent out to all devices on the list. This is detailed in Section 7.10.2.

8.1.2.1.3 Device Dropped From Configuration

When a Subordinate fails to respond to three consecutive Messages sent to its address with an ACK or other Message (this may include R2Rs, Messages from the Coordinator, or Routed Messages from a Subordinate), it is treated as dropped. The entry for this device is removed from the Node List and the updated Node List is sent to all remaining devices contained in the Node List. The address previously occupied by the dropped node also is removed from the list of addresses being serviced with R2Rs.

8.1.2.2 Configuration Tasker

The Coordinator is responsible for maintaining a list of all addressed nodes, their MAC addresses, and session IDs. When a new node is added to the list by the AutoNet Server, as specified in 11.11, the new node is declared Pending, after which an R2R is sent to it. This Message is called the Authentication R2R and is used to confirm that the node is present at that address. The Configuration Tasker is invoked after the presence at that address has been verified via the Response to the R2R.

The purpose of this Tasker is to authenticate and validate the new node before declaring it Active. This is done by sending a Node ID Request to the node. The node then responds with its Node Type, Session ID, and MAC Address. If these three items match the information the Coordinator already has, the node is considered Authenticated. Because the purpose of the Node ID Request is to authenticate newly addressed nodes, this Message should be the first addressable transaction to the node and shall immediately follow the Authentication R2R. Please consult the *Command Reference* for more information.

8.1.2.3 Network Node List Example and Framing for CT2.0 devices

The Network Node List contains a sequence of Node Types that, when the value is not zero (0), indicates the presence of a node at the physical address equal to the Node List Index with the type indicated by the non-zero value. Index 0 is a special reserved index for the optional presence of a Virtual Internal Subordinate within the Coordinator's subsystem. The maximum index in the Node List is equal to the highest supported Subordinate Address plus a one-byte Overhead Headroom Valuator that should be maintained as zero (0).

Below is an example of the payload data for a Network Node List for a network containing:

a. A Virtual Subordinate of Node Type 3 with a Node List Index of 0
b. A Priority Subordinate of Node Type 1 at a Physical Address of 1 (on Subnet 2 or Subnet 3) with a Node List Index of 1
c. A CT1.0 Subordinate of Node Type 5 at a Physical Address of 2 with a Node List Index of 2
d. A CT2.0 Subordinate of Node Type 24 at a Physical Address of 16 with a Node List Index of 16
e. A CT2.0 Subordinate of Node Type 24 at a Physical Address of 17 with a Node List Index of 17
f. A CT2.0 Subordinate of Node Type 1 at a Physical Address of 18 with a Node List Index of 18

Table 4 – Example Node List with a Virtual Internal Subordinate

Index	0	1	2	3	4	5	…	16	17	18	19	…	63
Node Type	3	1	5	0	0	0	…	24	24	1	0	…	0

The example below shows a network with no Virtual Subordinate and the following nodes:

a. A Priority Subordinate of Node Type 1 at a Physical Address of 1 with a Node List Index 1
b. A Non-Priority CT1.0 Subordinate of Node Type 4 at a Physical Address of 2 with a Node List Index of 2
c. A Non-Priority CT1.0 Subordinate of Node Type 2 at a Physical Address of 5 with a Node List index 5

Table 5 – Example Node List with no Virtual Internal Subordinate

Index	0	1	2	3	4	5	6	7	…	60	61	62	63
Node Type	0	1	4	0	0	2	0	0	…	0	0	0	0

8.1.2.4 "Condensed" Network Node List Example and Framing for CT1.0 devices

A special consideration is required when sending a Network Node list to CT1.0 devices. The rule regarding receipt of node lists up to a length of 240 bytes was in place for CT1.0 devices. There was, however, conflicting documentation in the ClimateTalk Command Reference V1.3 which stated: *"The Network Node List will always contain 16 bytes in the data payload and these will represent the device Node Types of the Virtual Subordinate application within the Coordinator and those of nodes at addresses 1 through 15*."

To ensure CT1.0 devices will detect the presence of CT2.0 devices in a node list, a special "condensed" node list shall to be sent to CT1.0 devices with all nodes identified in the first 16 bytes.

Node list index 0 is still reserved for the Network Coordinator's Virtual Internal Subordinate. Node list index 1 is still reserved for the priority subordinate. Remaining node list indexes 2 through 14 will be filled with a single instance of additional addressed network devices.

The rules for creating the "condensed" network node list for CT1.0 devices are as follows:

- Scan through all addressed CT1.0 devices and only allow the first instance of a node type to remain in the node list. Replace duplicated CT1.0 devices with a zero.
- Scan through all addressed CT2.0 devices. For each new device type located, fill in the open locations in the CT1.0 "condensed" node list with that node type. Only one indication of a node type is allowed in the "condensed" node list.
- Node lists sent to CT1.0 devices shall have a maximum data payload length of 16 bytes.

Below is an example of the payload data sent to a CT1.0 device for the example noted in Table 4

Table 6 – Example Node List sent to CT1.0 devices

Index	0	1	2	3	4	5	…	15
Node Type	3	1	5	24	0	0	…	0

8.1.3 Network Routing Coordinator Application (Task)

The Network Routing Coordinator Application is responsible for maintaining an active network configuration and routing tables based on that configuration. One of the Coordinator's tasks is to provide the Network Subordinates with a method for sending packets to each other. This is done by tracking Subordinate devices as they come onto and off of the network. Several routing tables are maintained based on the particular network configuration.

Any time there is a change to the network configuration, the Coordinator adjusts the routing tables according to each table's specific purpose.

All Subordinate devices are notified of the change to the network configuration. The Coordinator does this before normal network traffic is allowed to resume. Since Subordinates are only told of changes to the network after the routing tables are updated and Subordinates are not allowed to send traffic until the routing tables have been updated, the Subordinates are able to adjust to changes in the network more efficiently.

This Coordinator keeps all Subordinates refreshed with the current network configuration during its usual network maintenance by sending Network Node List Request Messages.

8.1.4 Network Shared Data Application

The Network Shared Data Application is responsible for maintaining a non-volatile, RAID-like, Network Shared Data Storage Repository system for Subordinate Applications.

Data is sent and retrieved by User Applications through requests to the Network Coordinator. Only certain Applications, as defined by their Application Profile, are allowed to send shared data to the Coordinator for storage and distribution.

Any device not specifically required to offer the storage of shared data in their Application Profile, may optionally provide the service to store Network Shared Data when so requested by the coordinator.

8.1.4.1 Network Shared Data Sectors

There are only three sectors, each containing a network Shared Data Record of up to two hundred bytes in length. These sectors are:

a. Sector 0 – Thermostat and Zone Controller (Node Type 1 and Node Type 21)
b. Sector 1 – Indoor Unit (Node Types 2 and 3)
c. Sector 2 – Outdoor Unit (Node Types 4, 5, and 12)

All ClimateTalk devices that are capable of becoming a coordinator shall provide the service of storing Shared Data for devices that make the storage request. A coordinator capable device shall distribute any Shared Data it receives to other devices on the network.

Each device with an application requirement to accept shared data or one that optionally provides shared data service shall store all three sectors and allow for their retrieval by the Coordinator. The device must track whether each sector has been written with shared data. Requests for shared data from a shared data sector that has not been populated shall respond with a payload length of zero.

Sectors 1 and 2 are stored and retrieved on a ‘last pushed data set stays’ basis, i.e. if an Air Conditioner pushes data to the Coordinator after a Heat Pump, the Heat Pump data is overwritten by the Air Conditioner data since they both share the same sector. All nodes assigned to these Sectors enjoy equal priority. Similarly, when the Coordinator pushes the Air Conditioner data out to Subordinates for replication, this replaces the previously stored data in that sector regardless to which Node Type it belongs. The same process of data replacement happens for new Shared Data pushed to the Coordinator by any nodes using Sectors 1 or 2. The new data replaces the old data.

Sector 0 offers an enhanced priority for the Zone Controller Node Type, i.e. any Zone Controller Shared Data pushed by the Application to the Coordinator or from the Coordinator to a Subordinate for replication overwrites the previously stored data in that sector. However, if Thermostat data is pushed and the Sector is already storing Zone Controller data, the push is ignored. To maintain backward compatibility, the push is responded to as if the data were accepted and stored.

8.1.4.2 Network Shared Data Taskers

A Push Tasker and a Pull Tasker running on the Coordinator facilitate redundant network storage of Shared Data. The Push Tasker makes sure all network devices on the network are storing the same and most up-to-date data. The Pull Tasker retrieves Shared Data for an application by requesting it from the redundant network storage. Both the network Push and Pull functions are accomplished through the Network Shared Data Sector Image Message, the details of which may be found in the applicable API reference.

8.1.4.2.1 Network Pull Tasker

When the Network Coordinator receives a Get Application Shared Data from Network message, it must serve up the Shared Data record it has stored in the sector corresponding to the requesting subsystem's Node Type. If the Network Coordinator does not have a valid record for the sector of the requesting node, then the Coordinator begins polling the other nodes in the network sequentially by address for the record. When the Pull Tasker reaches the last addressed node, it starts over from the first address. The Tasker continues until the desired record is found. The application must keep issuing the Request until it gets data to ensure the application gets the data once the Pull Tasker succeeds.

While the Coordinator is performing the Pull Tasker or if it has completed the pull Tasker and finds no data for that sector, it shall respond to the requesting application with an empty payload indicating no Shared Data is available for that sector. Similarly, if a Subordinate being queried for Shared Data has no valid data for the sector being queried, it shall return an empty payload to the Coordinator. In addition, since the only way Shared Data can get to a sector on a Subordinate is by a push from the Coordinator, if a given Subordinates has been queried for a given sector during the current session, the Coordinator does not need to query that Subordinate again for the same sector unless the session has changed.

More details on the application rules for Shared Data on the network are contained in Section 10.0. Details on the packet structure of the *Get Application Shared Data from Network* Message are in the applicable API reference.

8.1.4.2.2 Network Push Tasker

Any time the Network Coordinator receives a new Shared Data record through a Set Application Shared Data to Network message, the Network Coordinator must store the record, mark it as Valid, and then begin the Push Tasker. The Push Tasker pushes the newly stored record to every addressed node in order by address and then stops. This ensures every node has the most up-to-date data for the sector corresponding to the node participating in the push. There is no expectation for the amount of time it takes for this Tasker to completely push each record to every applicable node. This is the lowest priority task and is based on the actions of the application and other Taskers. The Push Tasker should complete within thirty minutes.

Certain rules exist for the application to make sure all up-to-date data is pushed out. Consult Section 10.0 for more information on these rules. Details on the packet structure of the Set Application Shared Data to Network Message are contained in the applicable API reference.

8.1.5 Subordinate Applications

Subordinate Applications serve as the interface to the end user. The user is defined as the system that will use the functionality provided by the Network Coordinator.

8.1.5.1 Application Node Types

Subordinate Applications are distinguished by a Node Type. For a list of common Node Types used in ClimateTalk, consult the Appendix of the *Command Reference*.

Certain Send Methods defined in section 6.3 do not support routing to multiple instances of the same node type. It is recommended to only have one instance of a node type noted in Table 3 – Priority Device Routing unless all applications on a network support send methods for multiple instances of that same node type.

8.1.5.2 Network Shared Data Push and Pull Support

All FFD nodes and all HVAC RFD nodes must support storage and the retrieval of three Shared Data sectors initiated by the Network Coordinator using the Network Shared Data Sector Image Message. Refer to Section 8.1.4.1 for details on the supported sectors.

During a Shared Data Sector Image **Read** Request, if no shared data is available or if the device does not provide the service to store shared data, the response shall return with a Payload length of 0x00. During a Shared Data Sector Image **Write** Request, if the device does not provide the service to store shared data, the response shall be returned with an indication of an Unknown Application Payload per the *Generic Application Specification*.

8.1.5.3 General Network User Application

Network User Applications are the real communicating devices in a ClimateTalk system. The rest of the system is in place to serve and support these Network User Applications. All Network User Applications are linked to a Subordinate device on the network.

Devices operating as a Coordinator can also run simultaneously as a Subordinate device. A Coordinator is allowed to have one (1) internally linked Subordinate device. The Coordinator reports this device as an Index of 0 in each Node List that is sent to a Subordinate.

9.0 Network Timing Requirements

9.1 General CT-485 Network Timings

This section discusses how to calculate the maximum bandwidth tolerance and determine the effect of the maximum strain conditions for the network.

To see the results at maximum tolerance to the network, all calculations in this section performed using the following assumptions:

a. The network is at maximum node capacity (all addresses in use)
b. All nodes take the maximum amount of allotted time to respond to each transaction
c. No transactions actually reach the timeout value associated with that transaction

To allow for a self-healing and dynamic Coordinator in CT-485, some specific timings must be followed for both the Network Coordinator and the Subordinates.

9.1.1 Coordinator Arbitration Timings

The determination of which FFD becomes Coordinator is decided by which FFD becomes an AutoNet Server. See Section 11.2 for details on Coordinator and AutoNet Server arbitration timings.

9.1.2 Address Dropping and Resolution Timings

To force a CT-485 V1.0 node to drop a previously assigned address or to resolve collisions at an address in Subnet 2, the Coordinator ignores that address by not sending any Directly Addressed Packets to that address for five minutes. This action occurs if the Response from a node at a given address does not match the MAC Address of the node that should be there or if collisions are detected whenever packets are sent to an address. When a CT-485 V1.0 Subordinate has not received a packet sent to its address for a period of two minutes, it reinitializes its presence on the network by renouncing its assigned address and subnet values. This allows a CT-485 V1.0 addressed node that has been ignored by the Coordinator to become readdressed through AutoNet. This also allows resolution of any scenarios where more than one Subordinate thinks they are at the same address.

There are, however, two methods to reinitialize CT-485 V2.0 nodes and to cause addresses to be dropped. The Coordinator shall send out an Address Confirmation Broadcast at least once every 120 seconds to all nodes on Subnet 3. All CT-485 V2.0 nodes shall watch for this packet. The nodes need to confirm that this Request has the same payload as the current Network Node List as understood by the Coordinator, ensure that the presence of the node resides at the Index corresponding to its own address, and that this entry matches its own Node type. At any time, if the Address Confirmation Broadcast is not seen for 120 seconds or its own Expected Node Position information does not match the Subordinate's expectations, the CT-485 V2.0 Subordinate shall relinquish its address and subnet values, re-enter the network, and wait to be addressed. The other method is to assign the values of zero (0) to the Subordinate's address and subnet.

9.2 Maximum Idle Network Cycle Time

Based upon the currently defined Cycling Sequence, the Idle Time Network Cycling Sequence is:

Dataflow Cycle Sequence	Maximum time (seconds)
R2R sent to Priority Subordinate	3.5 3.5 comes from 3 (maximum subordinate response time) + 0.5 (maximum coordinator turn-around time to send next message)
Node Discovery	3.5
Address Confirmation Broadcast	3.5
Token Offer Broadcast	3.5
R2R to each Subnet 2 address 0x01 through 0x0E	49 = 14 * 3.5
R2R to next addressed Subnet 3 node	3.5
R2R to Virtual Subordinate	3.5
Coordinator Transaction Time	3.5
	77.0 seconds

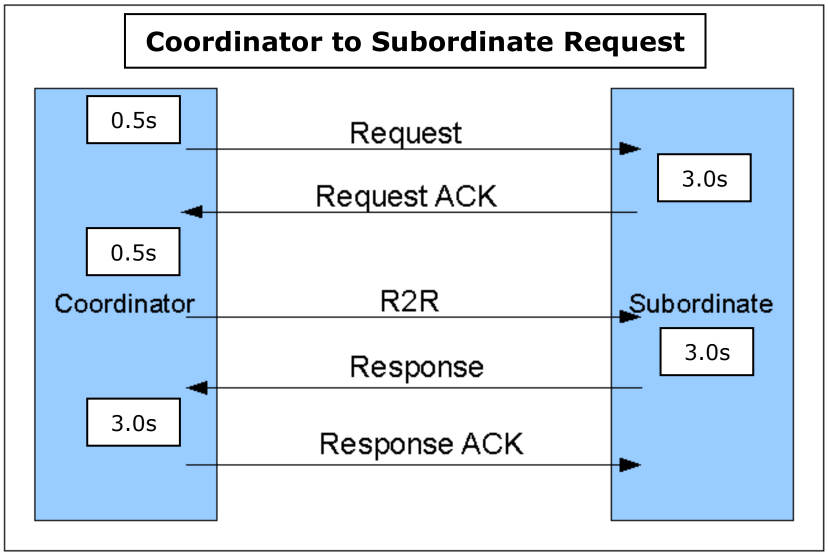
Given the assumptions of section 9.1 and assuming a Priority Subordinate is a CT1.0 device, the maximum Idle Network Cycle Time is 77.0 seconds.

9.3 Maximum Coordinator Initiated Transaction Timings

A Coordinator Initiated Transaction contains one Request Cycle, followed by one Request to Receive Cycle, followed by one Response Cycle. The Coordinator Initiated Transaction Request also contains two Coordinator Turnaround Times, one before the first Request Cycle and one before the Request to Receive Cycle:

Given the worst case timing assumptions of section 9.1, the maximum Coordinator to Subordinate Transaction Delay is 10.0 seconds.

Figure 8 – Maximum Coordinator to Subordinate Transaction Delay



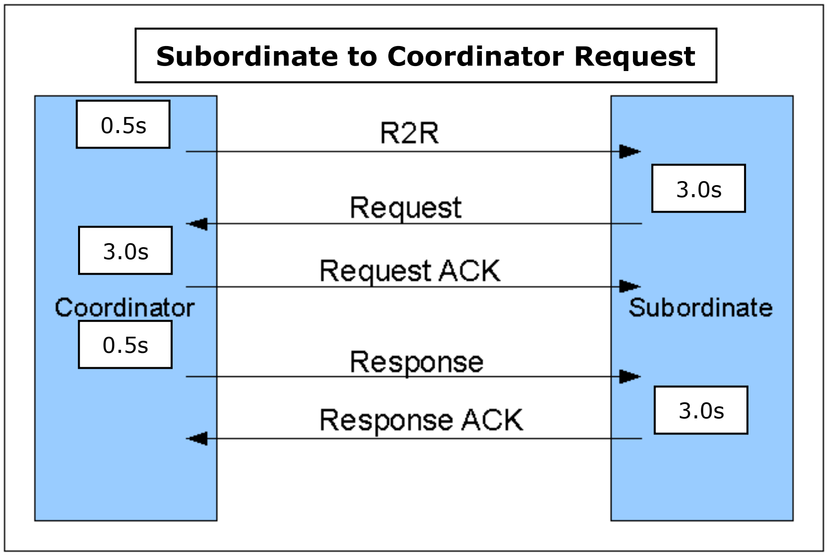
9.4 Maximum Subordinate Transaction Timings

9.4.1 Maximum Subordinate to Coordinator Transaction Delay

A Subordinate to Coordinator Request contains one Request to Receive Cycle followed by one Request Cycle followed by one Response Cycle. The Subordinate to Coordinator Request also contains two Coordinator Turnaround Times, one before the first Request Receive Cycle and one before the Response Cycle.

Given the worst case timing assumptions of section 9.1, the maximum Subordinate to Coordinator Network Transaction Delay is 10.0 seconds.

Figure 9 – Maximum Subordinate to Coordinator Transaction Delay

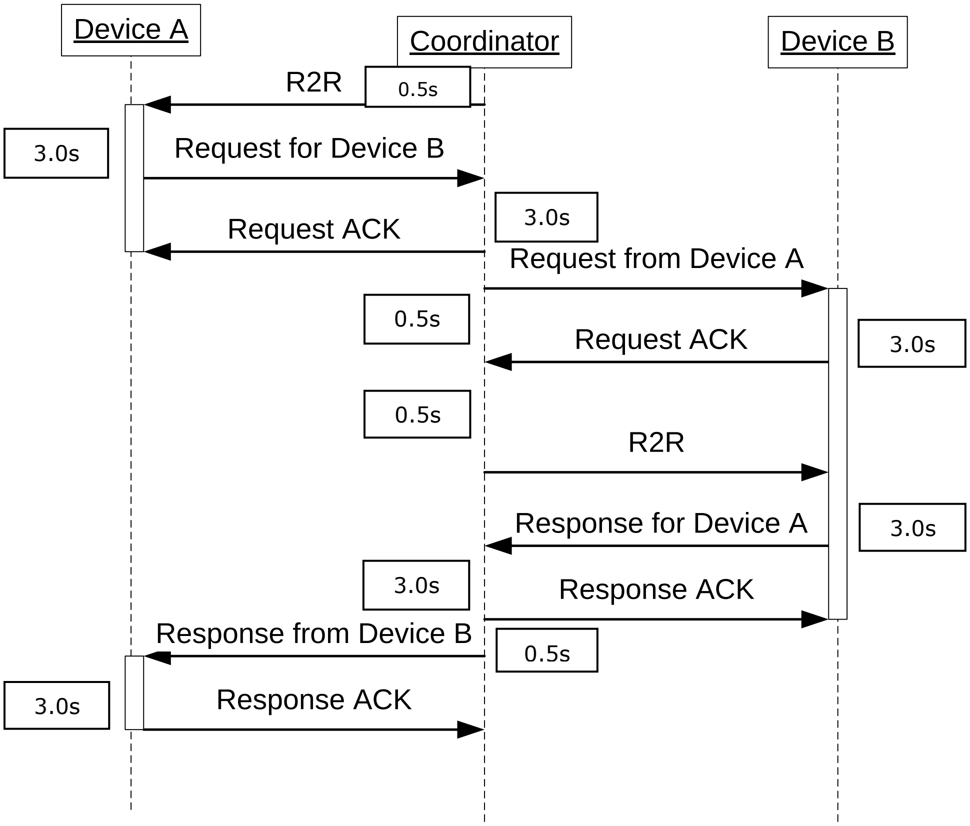


9.4.2 Maximum Subordinate to Subordinate Transaction Delay

A Subordinate to Subordinate Transaction consists of a Coordinator to Subordinate Transaction wrapped within a Subordinate to Coordinator Transaction. Therefore, the theoretical maximum time for a Subordinate to Subordinate Transaction is the sum of those two maximums. Each of which was determined in this section to be 10 seconds.

Given the worst case timing assumptions of section 9.1, the maximum Subordinate to Subordinate Network Transaction Delay is 20.0 seconds.

Figure 10 – Maximum Subordinate to Subordinate Transaction Delay



9.5 Response speed recommendation

Mandating the sending of an R2R for every CT-485 V1.0 node in every cycle means that every CT-485 V1.0 node gets a Transmission Opportunity at least every 120 seconds, regardless of bandwidth use.

Even though the CT-485 specification mandates a 120-second cycle, this may not be possible if all devices delay the maximum response times.

To improve the responsiveness of the network, it is recommended that the Subordinate always responds to messages as soon as possible while observing the minimum 100 millisecond bus delay and that the Coordinator also turns around as soon as possible while also observing the minimum 100 millisecond bus delay.

10.0 Shared Data

10.1 Shared Data Definition

Shared Data consists of non-safety related parameters that can be programmed on the manufacturing line to tailor the operation of the control. This data is communicated serially to the control during OEM test mode on the OEM's manufacturing line. The Shared Data record for each subsystem may be up to two hundred bytes in length.

Shared Data records are stored redundantly by both the application and the network. Records accepted by the application are pushed to the Network Coordinator and then copied to each device on the network. This helps ensure the survival of mission critical data for a control, even in the case of a replacement board that has no Shared Data.

It is the responsibility of the network to ensure that each device's Shared Data is distributed to the appropriate backup devices on the network.

For information on the application requirements for Shared Data, see the section on Shared Data in the *Generic Application Specification*.

10.1.1Request Shared Data

From a cold start, the subsystem shall determine whether it has valid Shared Data from any of the possible Shared Data sources other than the network. If the subsystem does not possess valid Shared Data, it shall request Shared Data once it is active on the network. The subsystem shall continue to request Shared Data from the network until it acquires valid Shared Data.

The application requests Shared Data from the Network Coordinator using the Get Application Shared Data from Network Message. The Network Coordinator requests Shared Data from the other devices on the network using the Network Shared Data Sector Image Read/Write Message.

10.1.2Push Shared Data

On receipt of a new Network Node List, the subsystem shall push out its Shared Data to the network if it is in possession of valid Shared Data. If the subsystem is active on the network without valid data, it shall push out Shared Data as soon as it is in possession of valid Shared Data. In addition, the subsystem shall push Shared Data out to the network if any changes are made to the Shared Data stored on the subsystem while the subsystem is active on the network.

Shared Data is pushed out by the application using the Set Application Shared Data to Network Message. The Network Coordinator sends the Shared Data to the other devices on the network using the Network Shared Data Sector Image Read/Write message.

11.0 AutoNet over Serial Physical Layer

AutoNet is a pseudorandom bus-addressing algorithm that uses a DHCP-Lite protocol, which allows automatic addressability of CT-485 Half Duplex Master/Slave Addressable Network Modules.

AutoNet is a proprietary embedded implementation of the Dynamic Host Configuration Protocol (DHCP).

The following specialized functionality is added when using CT-485 as the communications protocol to implement AutoNet:

a. Determining the Network Coordinator device for the CT-485 network
b. Sharing the control of the CT-485 serial bus with the Network Coordinator device
c. Controlling and disabling AutoNet for CT-485 systems via packet manipulation
d. Maintaining MAC Addresses and Sessions

The concept of the algorithm is to use a series of real-world and real-time events to synchronize a specific network node as the AutoNet Server and then communicating with multiple network nodes that are selectively and verifiably addressed as AutoNet Clients one at a time.

The algorithm takes advantage of several system-timing differences between network subsystems including discrete temperature profiles and silicon production variances that occur based on statistical probability and the physics of semiconductor production.

11.1 Slot Delay

A Slot Delay is a pseudorandom number that indicates a time delay with a range from one hundred milliseconds to twenty-five hundred milliseconds with a granularity of one millisecond and a uniform [rectangular] distribution over the entire range. All devices must be capable of generating a Slot Delay that adheres to the above requirements. The pseudorandom number algorithm generating the Slot Delay also must be externally seeded so that identical devices on the network do not end up with the same Slot Delays.

11.2 AutoNet Server

The AutoNet Server is the internal system responsible for addressing new nodes on a CT-485 network. New devices are polled regularly so that they can announce their presence. If a new device is found, the AutoNet Server is responsible for giving that device an address and subnet for use on the CT-485 network.

There can be only one AutoNet Sever in a CT-485 system and that AutoNet server is symbolically linked to the Network Coordinator. In other words, if a device becomes the AutoNet Server in a system, it also becomes the Network Coordinator for that system.

11.3 AutoNet Clients

AutoNet Clients are devices that have determined via their internal system that they are not an AutoNet Server. AutoNet Clients must wait for an AutoNet Server to poll for new nodes before being allowed to announce their presence. Once the AutoNet Server has been notified of a new client device, the client is given an address and subnet to use for the current network session.

All devices start out as AutoNet Clients. If it is determined that there is no AutoNet Server present, then all capable AutoNet Clients arbitrate to become the AutoNet Server. This type of arbitration makes sure that there is always an active AutoNet Server (and Network Coordinator) in a system.

11.4 Coordinator Arbitration

CT-485 utilizes a method of Coordinator Arbitration to determine which FFD on the network will serve as the AutoNet Server and Coordinator for the session. This method uses Slot Delay (discussed below) after a period of listening to give each device a unique time delay before attempting to become the AutoNet Server as well as a version announcement and arbitration process known as Coordinator Arbitration Version Announcement (CAVA).

Coordinator Arbitration only applies to devices that are capable of becoming a coordinator. A device not capable of becoming a coordinator shall refrain from transmitting a Coordinator Arbitration Version Announcement message in reply to a node discovery message. A device not capable of becoming a coordinator shall refrain from transmitting a Coordinator Arbitration Version Announcement message in reply to a broadcast Coordinator Arbitration Version Announcement message.

At any time, due to initial power up or prolonged silence on the bus (over 120 seconds), the following Coordinator Arbitration procedure begins:

```
START: Wait for a period greater than six seconds but less than thirty seconds and listen
on the network. Is any traffic seen?
1. Yes. Wait for CAVA or Node discovery request
    1.1. CAVA seen. Is own CAVA > received CAVA?
        1.1.1. No– Go quiet. If Internal Subordinate is present, come up as Subordinate
               to get on network.
        1.1.2. Yes – Start Slot Delay. Is traffic seen within Slot Delay?
            1.1.2.1. Yes – CAVA. Repeat at Step 1.1
            1.1.2.2. Yes – Other traffic. Go to Step 1
            1.1.2.3. No – Send own CAVA. Wait for response until timeout. Response?
                1.1.2.3.1.Yes – Repeat at Step 1.1
                1.1.2.3.2.No – Become the Coordinator
    1.2. Node Discovery Request seen?  Start Slot Delay.  Any packet seen within Slot
         Delay?
        1.2.1.  Yes – CAVA seen. Repeat at Step 1.1
        1.2.2.  Yes – Node Discovery Response seen? Go to Step 1
        1.2.3.  No – Go to Step 1.1.2.3
2.  No. Go to Step 1.1.2.
```

Conversely, the existing Network Coordinator shall hand over control to a CAVA process any time a CAVA is observed on the network, regardless of what the Coordinator was doing when CAVA was received. The existing Network Coordinator check is as follows:

```
1.  CAVA seen. Is own CAVA ≥ received CAVA?
    1.1. No– Go quiet.  If Internal Subordinate is present, come up as Subordinate to get
         on network.
    1.2. Yes – Start Slot Delay.  Is traffic seen within Slot Delay?
        1.2.1.No – Send own CAVA. Wait for response until timeout.  Response?
            1.2.1.1.Yes (has to be CAVA)– Repeat Step 1.1
            1.2.1.2.No – Continue being the Coordinator
        1.2.2.Yes (has to be CAVA)– Repeat Step 1.1
```

The existing Coordinator will respond to CAVA even when its own version is only equal to the observed CAVA whereas a new FFD in arbitration will only attempt to become Coordinator if its CAVA is better than the observed CAVA. This promotes stability on the network by not changing an existing Network Coordinator unless another FFD is of a higher version.

An existing Network Coordinator will respond with equal CAVA only once. If it sees a CAVA from another FFD that is equal to its own again, it will yield and accept defeat in the arbitration process.

All CAVA messages, regardless of whether they are sent by the current Network Coordinator or by a new FFD trying to become the Network Coordinator, shall be sent with the source address set to 0xFF, the destination address set to 0xFE, and the Subnet set to zero. Further, all FFDs, whether currently the NC or not, shall always listen to, process, and respond to (if appropriate per above rules) CAVA messages received with the destination address set to 0xFE and the Subnet set to their own Subnet or the Broadcast Subnet. The address 0xFE shall be treated as a special Broadcast address reserved for Coordinator Arbitration purposes.

In the above procedure, any CAVA comparison is as follows. The fields included in the payload of the CAVA message are:

a. FFD Capable – enabled / disabled
b. CT-485 Version – unsigned 16-bit value

c. CT-485 Revision – unsigned 16-bit value

When FFD A sees CAVA from FFD B, one of the following conditions has to be satisfied for FFD A to cue up a response with its own CAVA. In other words, the result of TRUE for the check of (own CAVA is greater than received CAVA) in the above process comes from a true result for one of the following conditions:

a. CT-485 $\text{Version}_{\mathbf{A}}$ > CT-485 $\text{Version}_{\mathbf{B}}$
b. (CT-485 $\text{Version}_{\mathbf{A}}$ = CT-485 $\text{Version}_{\mathbf{B}}$) and (CT-485 $\text{Revision}_{\mathbf{A}}$ >CT-485 $\text{Revision}_{\mathbf{B}}$)

At the end of the process above, the device will either have become Coordinator or would have yielded to either go quiet if no internal Subordinate is present or enter the network as a Subordinate if one is present internally.

Once an FFD wins Coordinator Arbitration, it must broadcast a Network State Request to Subnet 3 and wait until timeout for a response. If a response is received, the payload will mirror the last network node list that existed on the network. The new Coordinator must now send a Node ID Request to each address indicated by the network state observed and, if an acknowledgement is seen, complete the transaction. All addresses except 1 will be unambiguous for this step since addresses 0x02-0x0E will be occupied on Subnet 2 and addresses 0x10-0x3E only on Subnet 3. If the network state shows Node List position 1 was occupied, the new NC shall send a Node ID Request first to Subnet 2 and, if no acknowledgement is seen, to Subnet 3. For every device whose presence has been confirmed, the Coordinator shall add it to its configuration, but shall not yet send out Network Node List Update Notifications. Once all already-addressed devices have been captured, the Coordinator shall send out a Node List to all discovered devices and proceed to perform its first Dataflow Cycle.

11.5 Coordinator Re-Arbitration

If an FFD becomes a Subordinate or goes quiet on the network as a result of Coordinator Arbitration and there is no traffic on the network for one hundred and twenty seconds, it shall re-enter the Coordinator Arbitration procedure. This starts with the listening period and ends with a Slot Delay. One of the FFD devices becomes the Network Coordinator again as specified in Section 11.4.

11.6 Coordinator Yielding

If the current Coordinator sees traffic on the network that appears to emanate from another node on the network believing itself to be the Coordinator, it shall re-enter the Coordinator Arbitration procedure starting with the initial Listening Period and ending with a slot delay and one of the FFD devices becoming coordinator again, as specified in Section 11.4.

11.7 Media Access Control and Sharing

CT-485, in its most basic form, places significant limitations on a communications system. One of the most significant limitations is that, in order to support unidirectional CT-485, only one device can be driving the physical media at any given time.

In a CT-485 system, the Coordinator acts as a Network Controller Device and is allowed to send traffic during any idle time on the network. Subordinates must wait until they are given time to send their traffic by the Coordinator. Therefore, only one device can send data at any given point in time.

AutoNet uses a similar system of control. An AutoNet Server acts in a manner similar to a Network Coordinator and an AutoNet Client acts similarly to a Network Subordinate.

The solution for centrally controlling a device on a CT-485 bus is to allow the CT-485 network services to share the bus with the AutoNet services and by logically linking the Coordinator and AutoNet Server together within the internal system.

Since the AutoNet Server and the Network Coordinator both control the CT-485 system during idle time, they are linked together in the software of each CT-485 device. If a device becomes the AutoNet Server (based on arbitration), that device also assumes Network Coordinator duties for the network. This allows the system to share the idle time on the bus more efficiently by allowing one device to arbitrate the sharing of the physical CT-485 connection internally without disrupting the communication media.

11.8 CT-485 Network Sessions

A CT-485 Session starts whenever a new AutoNet Server takes control of a system and ends when that AutoNet Server releases control of the system. Since the AutoNet Server within a CT-485 system is also the Network Coordinator, a session also can be viewed as starting when a new Network Coordinator takes control on a system and ending when the Network Coordinator releases control.

11.9 Session ID Allocation

Each time a new AutoNet device reboots or restarts, it generates a new pseudorandom Session ID for itself. This Session ID is included in all dataflow packets [R2R and ACK] sent from that device. Changes to the reported Session ID indicate that a network device has been rebooted or restarted and needs to be re-configured for use on the CT-485 network.

The pseudorandom number algorithm for generating the Session ID has to follow a range from one (1) to $2^{64} - 2$ with a granularity of one. The pseudorandom Session ID generation algorithm also must be seeded externally so that identical nodes do not generate the same Session ID.

Handling Session ID this way allows AutoNet to be independent of the higher-level network functions and still allows the higher-level network functions to react to changes in the devices on the network.

11.10 AutoNet Subnet Methodology

The AutoNet Server provides the Coordinator a service by:

a. Discovering unaddressed nodes and assigning them an appropriate unoccupied address suited to their Node Type
b. Fetching the information about their MAC addresses, Session IDs, and Node Types

c. Filing the information into a Subordinate Operational State Database maintained by the Coordinator enabling the coordinator to add newly discovered nodes to its session
d. Providing an updated Node List to all nodes, including newly discovered nodes

The AutoNet Server uses the complement of the occupied addresses from the Coordinator's Subordinate Operational State Database as its universe of potentially assigned addresses. If, in the course of interrogating addresses believed to be unoccupied, one of these addresses actually is occupied, the AutoNet Server is responsible for confirming the address and fetching the information about the occupant node's MAC Address, Session ID, and Node Type. It then fills them into the Coordinator's database and informs the Coordinator so that the Coordinator can, once again, add the recovered node to its session and provide an updated Node List to all nodes, including the newcomer.

Consult the CT-485 API for packet details regarding Node Discovery, Set Address Request, and Set Address Response Messages used in the following subsections.

11.11 AutoNet Server Procedures

The following is a systematic description of what is performed by a node when it becomes an AutoNet Server and begins to address nodes using AutoNet.

1. Send a Node Discovery Request
2. If a response is received, determine the type of response
   2.1. Node Discovery Response:
   2.1.1. Temporarily store the MAC Address and Session ID
   2.1.2. Determine target Subnet
      2.1.2.1. Check the 'VERSION' field in the packet number
         2.1.2.1.1. If node is CT-485 V1.0 (VERSION field = 1), target Subnet is '2'
         2.1.2.1.2. If node is not CT-485 V1.0 (VERSION field = 0), target Subnet is '3'
   2.1.3. Determine target address
      2.1.3.1. If discovered node is a Thermostat or Zone Controller type
         2.1.3.1.1.If address '1' has not been assigned,
            2.1.3.1.1.1.Target address is '1'
      2.1.3.2. Otherwise, target address is next empty address on target Subnet
         2.1.3.2.1. If target Subnet is 2, target address is next empty address in the range 0x02-0x0E. If no target address is available, abort the AutoNet process.
         2.1.3.2.2. If target Subnet is 3, target address is next empty address in the range 0x10-0x3E. If no target address is available, abort the AutoNet process.

2.1.3.2.3. If a CT1.0 device is on the network or if the discovered node is a CT1.0 device, abort the AutoNet process if there are no available node list positions for the discovered node in the "condensed" node list. Reference paragraph 8.1.2.4.

2.1.4.Send Node ID Request to target assignment address on target Subnet

2.1.5. Wait three seconds for acknowledgement

2.1.5.1. If an acknowledgement is received

2.1.5.1.1. Complete transaction to get node ID response

2.1.5.1.2. Verify that the device is in a node-appropriate address in a version-appropriate Subnet

2.1.5.1.2.1.Yes – Device's address and Subnet are fine

2.1.5.1.2.1.1. Add device to configuration and inform Coordinator Application so it can send out Node Lists to all nodes including the newly discovered node

2.1.5.1.2.1.2. Move to the next unoccupied address on the Subnet and repeat Step 2.1.4 to address the discovered device

2.1.5.1.2.2. No – Device does not belong at this address / Subnet due to Node Type or CT-485 version

2.1.5.1.2.2.1. Send a Set Address message with Address 0 and Subnet 0 to the device now occupying the address in order to pick it up in the next round of node discovery. If acknowledgement is seen, complete transaction

2.1.5.1.2.2.2. Lock out this address (i.e. send no traffic to) for two minutes and fifteen seconds to allow the node to lose the address

2.1.5.1.2.2.3. Move to the next unoccupied address on the Subnet and repeat Step 2.1.4 to address the discovered device

2.1.5.2. If no response is seen, send a Set Address message using the stored MAC Address, Session ID, and Assignment Address on the target Subnet with the Write Enabled set to 0x01

2.1.5.3. Wait three seconds for a Set Address Reply

2.1.5.3.1. If a Set Address Reply is received from the new address that the AutoNet Server assigned

2.1.5.3.1.1. Add device to configuration and inform Coordinator Application so it can send out Node Lists to all nodes including the newly discovered node

2.2. Coordinator Arbitration Version Announcement:

2.2.1. Hand control over to the Coordinator Application to manage Arbitration

3. Repeat Step 1 whenever the Coordinator Application requests a Node Discovery

During the above procedure, the AutoNet Server shall always make an effort to check whether a newly discovered node's MAC Address matches another device already believed to be present on the network. If this is the case, the AutoNet Server makes an attempt to provide the newly discovered node the same address it had occupied previously if the address is still empty.

11.12 AutoNet Client Procedures

The following is a systematic description of what is performed by a node when there is already an AutoNet Server on the network and, based on Coordinator Arbitration, has become a Subordinate. If an RFD, it is by definition determined to be a Subordinate, so it will always follow the procedure below when not addressed. Further, even though CT-485 V2.0 specifies address and Subnet ranges for the Coordinator to assign, every Subordinate must be capable of accepting any address and Subnet assigned to it to enhance compatibility with future versions of Network Coordinators.

1. Wait for a Node Discovery Request from the AutoNet Server
2. When received, generate a Session ID and a Slot Delay per Section 11.1
3. Set a timer to count down the newly generated Slot Delay and listen for traffic while waiting for the Slot Delay to expire
4. If no traffic is received before the Slot Delay Time expires

   4.1. If the Node Discovery Request filter value is equal to the Node Type or zero (0) (all Node Types), send a Node Discovery Reply using the current Session ID and MAC Address

   4.2. Wait for a Set Address Message that contains the current Session ID and MAC Address

   4.3. If a Set Address Message is received

      4.3.1.Check that the three conditions below are satisfied

         4.3.1.1. Check that the Session ID in the packet is the correct Session ID

         4.3.1.2. Check that the MAC Address in the packet is the correct MAC Address

         4.3.1.3. Check to see if the Write byte is Enabled (Write byte = 1)

      4.3.2. If all the conditions above are met for the incoming Set Address Message

         4.3.2.1. Send a Set Address ACK Message to the AutoNet Server

         4.3.2.2. Store Address and Subnet

11.13 Dataflow Cycles

CT-485 V2.0 specifically defines the mandatory cycling requirements to ensure the appropriate service levels. The steps defined below shall be performed at least once every 120 seconds. If time remains, the Coordinator may perform other required Coordinator initiated Transactions or start the next cycle early. The actual time taken to complete the cycle detailed below varies from cycle to cycle depending on how many nodes are present, how many nodes need to initiate Transactions, and how long each node takes to respond.

The following list also incorporates AutoNet Server functions since the Coordinator and the AutoNet Server are the same entity with the network time shared between them.

The Coordinator shall do the following operations at least once every 120 seconds:

1. Check to see if a priority subordinate node type is in the current node list?
    - 1.1. No - Send Node ID Request to Address 1 on each subnet
        - 1.1.1. Is a Response seen
            - 1.1.1.1. Yes
                - 1.1.1.1.1. Address the responder to appropriate subnet based on the corresponding CT version
                - 1.1.1.1.2. Send out Node Lists*
                - 1.1.1.1.3. Start a new session
                - 1.1.1.1.4. Restart the cycle
            - 1.1.1.2. No
                - 1.1.1.2.1. Continue cycle
    - 1.2. Yes - Send R2R to Address 1 on subnet where it is present
        - 1.2.1. Does it start a transaction
            - 1.2.1.1. Yes
                - 1.2.1.1.1. Complete Transaction
                - 1.2.1.1.2. Continue cycle
            - 1.2.1.2. No
                - 1.2.1.2.1. Continue cycle
2. Send Node Discovery Request
    - 2.1. Node Discovery Response seen
        - 2.1.1. Address the node to appropriate subnet
        - 2.1.2. Send out Node Lists*
        - 2.1.3. Start a new session
        - 2.1.4. Restart the cycle

2.2. Coordinator Arbitration Version Announcement seen

2.2.1. Start CAVA process

2.2.1.1. Win CAVA

2.2.1.1.1. Continue cycle

2.2.1.2. Lose CAVA

2.2.1.2.1. Go silent

2.2.1.2.2. No longer a Coordinator

2.3. No response seen

2.3.1. Continue cycle

3. Send Address Confirmation Broadcast to Subnet 3 if a node other than the virtual internal subordinate is addressed on subnet 3

4. Send Token Offer Broadcast to Subnet 3 if a node other than the virtual internal subordinate is addressed on subnet 3

4.1. Response seen

4.1.1. Send R2R to responder

4.1.1.1. ACK or timeout

4.1.1.1.1. Continue cycle

4.1.1.2. Message seen

4.1.1.2.1. Complete transaction

4.1.1.2.2. Repeat from 4. Send Token Offer Broadcast to Subnet 3 until a maximum of five Token Offer Broadcast messages have been sent during this dataflow cycle.

5. Send R2R to each addressed Subnet 2 node individually. For each R2R:

5.1. ACK or timeout

5.1.1. Continue cycle

5.2. Message seen

5.2.1. Complete Transaction

6. Send R2R to the next Subnet 3 node in a slow rolling list

7. Send R2R to Virtual Internal Subordinate

8. Any other Coordinator Transactions

** Node Lists are sent directly and individually to each Subnet 2 node, directly and individually to each newly added subnet 3 node, and may be broadcast to all other Subnet 3 nodes. A broadcast node list to subnet 3 is required to ensure a node that considers itself as being addressed, will receive an indication the coordinator no longer considers that node part of the network.*

11.14 Transaction Timeout and Invalid State Rules

During the cycle, if any non-ACK packet to a Subordinate times out, the Coordinator performs the following actions in attempt to complete the transaction that was attempted:

1. R2R looking for a Request - If an R2R to a Subordinate looking for it to send a Request times out, the Coordinator provides it with R2Rs up to two more times or until it responds with a CT-485 packet of any kind (Request, Response, or ACK), whichever happens first
   1.1. If the two extra R2Rs also time out, the Coordinator moves on to the next step in the current Dataflow Cycle and the node is flagged for a Subordinate Presence Confirmation to be performed at the end of the current cycle
2. Follow-up R2R looking for Response - If it is a follow up R2R being sent to a node checking for any Responses to a Request previously delivered, the Coordinator provides it with up to two more R2Rs or until it responds, whichever happens first
   2.1. If the two extra R2Rs also time out, the Transaction is abandoned, the Coordinator moves on to the next step in the current Cycle, and the node is flagged for a Subordinate Presence Confirmation, which is performed at the end of the current cycle
3. Request Message - If a Request Message timed out with no ACK, the Coordinator resends the original Message up to two more times, or until it receives an ACK, whichever happens first
   3.1. If the two extra attempts also time out, the Coordinator moves on to provide the Receiving Node with follow-up R2Rs to attempt to complete the current Transaction subject to the rules for the timeout of a follow-up R2R as specified above and the node is flagged for a Subordinate Presence Confirmation, to be performed at the end of the current cycle
4. Response Message - If a Response Message timed out with no ACK, the Coordinator resends the original Response Message up to two more times or until it receives an ACK, whichever comes first
   4.1. If the two extra attempts also time out, the Coordinator treats the current Transaction as complete, moves on to the next step in the current Dataflow Cycle, and the node is flagged for a Subordinate Presence Confirmation, to be performed at the end of the current cycle

The Subordinate Presence Confirmation sequence outlined above rules is detailed in Section 11.16. In addition to this fault-tolerant communication operation, the Coordinator shall abide by the following rules for completing any transactions:

a. If a Response Message is received by the Coordinator when it is not expecting a Response to itself or to be routed to another Subordinate, the response shall be discarded
b. If an ACK is received by the Coordinator in response to a follow up R2R looking for a Response, the received ACK is treated the same as a timeout according to the rules above
c. The Coordinator shall manage and handle only one active transaction at a time, which shall be completed or discarded before moving on to another

11.15 Dynamic Address Assignment Theory

Once the AutoNet Server receives a successful receipt of a *Node Discovery Reply* command (meaning that the checksum and packet integrity is validated as good), then the AutoNet Server temporarily stores the MAC Address, Session ID, and Node Type. This information is stored and used within the following command sequence sent to the AutoNet Client, which is the *Set Address* command.

By using unique data that was provided by the AutoNet Client node (i.e. MAC Address, Session ID, and Node Type) to set the AutoNet Client's logical address, the probability is high that the dynamic addressing will selectively allow one-and-only-one node to be logically addressed in this step. Furthermore, the AutoNet Client is guaranteed to be addressed correctly as long as the MAC Address was guaranteed to be unique in manufacturing.

However, in case of a manufacturing process challenged device, the protocol shall allow a pseudorandom MAC address to be generated by the device once in its lifetime, which is stored in non-volatile memory and used forever. The restriction is that the highest byte not be zero (0). MAC Addresses assigned by a Manufacturer have zero (0) for the highest byte. This ensures that a pseudorandom MAC Address does not collide with a genuine programmed MAC address. Thus, a pseudorandom range from 2^56 to 2^64 with a granularity of one and a uniform distribution over the entire range is allowed. This method is expected to be used by replacement devices.

Based on this allowance, the probability of errors in dynamic addressing is approximately one in x, where x is equal to approximately:

x ~ (2^128) [independent of Node Type]
or
x ~ (2^136) [when using specified Node Type addressing]

11.16 Subordinate Presence Confirmation

When any Subordinate does not send an ACK or other Message for a Directly Addressed Packet, the Coordinator shall initiate a Subordinate Presence Confirmation routine for that node after it completes the current Dataflow Cycle. The Subordinate Presence Confirmation routine consists of three consecutive R2R packets sent to the node that failed to respond to the Message.

If the node fails to respond to all three R2R packets, it is deemed to have dropped off the network. The Network Node List shall be updated and a new session shall be started.

12.0 Annex A – Bibliography

“TIA-485 (Revision A), Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems” *Telecommunications Industry Association*, 1998.

Zimmermann, Hubert (April 1980). "OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection". *IEEE Transactions on Communications*