A UDP/IP Adaptation of the ZigBee Application Protocol
draft-tolle-cap-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 11, 2009.

Abstract

This document describes a UDP/IP adaptation of the IEEE 802.15.4-based ZigBee Application Protocol that enables IP hosts to communicate using the application profiles and data models described by that protocol, over a wide range of links. This modified version of the ZigBee Application Protocol is named CAP (Compact Application Protocol), and it is intended to provide a complete stack of application profiles, data exchange, binding operations, security protocols, and discovery to IP-networked hosts and embedded devices. The protocol’s domain of applicability includes IEEE 802.15.4-based 6LoWPAN devices, but also those on conventional wired and wireless links and emerging powerline communication networks.
Table of Contents

1. Introduction ................................................. 3
   1.1. Usage Model ........................................... 4
   1.2. Terms Used ............................................ 5
   1.3. Requirements notation ................................. 8
   1.4. ZigBee Notice ......................................... 8
2. Transport .................................................. 8
3. Bootstrapping ............................................... 8
4. Address Replacement ......................................... 9
5. Core Protocol ............................................... 11
   5.1. Protocol Behavior ...................................... 12
      5.1.1. Transmission ....................................... 12
      5.1.2. Reception .......................................... 12
6. Data Protocol ............................................... 13
7. Management Protocol ......................................... 14
   7.1. Discovery Messages ..................................... 15
   7.2. Binding Messages ....................................... 18
   7.3. Binding Table Cache Messages .......................... 20
   7.4. Network Management Messages ........................... 21
8. Security Protocol ........................................... 22
   8.1. Secure Communication .................................... 22
   8.2. Key Management Protocol ............................... 24
      8.2.1. Key Management Protocol Examples .................. 25
      8.2.2. Key Management Protocol Messages .................. 28
9. Security Considerations .................................... 30
10. IANA Considerations ....................................... 31
11. References ................................................ 31
   11.1. Normative References ................................... 31
   11.2. Informative References ............................... 32
Author’s Address ............................................. 32
Intellectual Property and Copyright Statements ................. 33
1. Introduction

Embedded networking technologies such as IEEE 802.15.4 [IEEE802154] and 6LoWPAN [RFC4944] have enabled a new class of embedded systems to operate with native IPv4 and IPv6 [RFC2460] connectivity. IP enables interoperability at the network layer, but does not address the desire among users for a common application-layer standard that enables administrators of networks including IP-connected embedded systems to easily connect devices produced by different vendors.

The ZigBee Application Layer [ZigBee], and the ZigBee Cluster Library [ZigBeeCL] that runs over it, are designed to address this desire for devices on a specific wireless link (IEEE 802.15.4). The ZAL and ZCL describe a set of structured communication primitives for exchange of data and commands, a protocol for binding and service discovery, a security protocol, and a profile system that enables the definition of interfaces for specific classes of devices that are designed to discover each other and interoperate. The primary design goal of the ZAL and ZCL appears to have been the definition of a compact and low-traffic message format in order to support embedded systems attached to low-bandwidth lossy networks. A secondary design goal of the ZAL and ZCL appears to have been for a protocol that can be implemented using a small amount of code and RAM, to support embedded systems running on microcontrollers. These design principles make the ZAL and ZCL particularly well-suited for structured communication among networked embedded systems for instrumentation, monitoring, and open-loop control.

However, the ZigBee Application Layer was originally designed to operate only over IEEE 802.15.4 wireless networks. The full ZigBee Specification includes the definition of a network layer that also operates only over IEEE 802.15.4 wireless networks, and the ZAL is defined solely in terms of this network layer and the addressing primitives provided by the 802.15.4 link layer. Furthermore, it does not address how such an application protocol is carried out over other well-established communication links, nor how a conventional TCP/IP or UDP/IP host can participate in such an application protocol.

This document proposes an adaptation of the ZigBee Application Layer that enables devices with native UDP/IP stacks to exchange application-layer data while employing the higher-level primitives defined by the ZAL. Our name for this adapted version of the ZAL, as used within this document, is CAP (Compact Application Protocol).

CAP is not a "tunneling" mechanism, by which multiple ZigBee networks may be interconnected over an IP substrate, in which ZigBee messages are passed unmodified. CAP is also not a "gatewaying" mechanism, by
which messages from ZigBee nodes may be modified by an intermediate host, possibly with address reassignment or payload modification, and placed into an IP message for transmission to other IP hosts.

CAP is a "port" of the ZigBee Application Layer from its original IEEE 802.15.4-specific network layer to a native UDP/IP implementation. ZAL messages are placed into UDP messages, and particular modifications to the ZAL protocol are required because the substrate is now IP hosts with IP addresses instead of IEEE 802.15.4 hosts with 802.15.4 addresses.

By running this "ported" ZAL service, IP hosts at all scales (embedded, mobile-class, PC-class, and server-class) can benefit from the data, binding, discovery, and profiles defined by the ZAL.

The designer of a host running CAP may choose to implement the Devices and Clusters of any of the published ZigBee Application Profiles over CAP, including but not limited to ZigBee Smart Energy [ZigBeeSE] and ZigBee Home Automation [ZigBeeHA]. The designer may also choose to implement any unpublished ZigBee Application Profiles, or define their own Application Profile for a new application domain.

Several other related application-layer standards exist for providing structured communication, binding, discovery, and profiles over IP networks, including Devices Profile for Web Services [DPWS] (and its underlying Web Services standards) and Universal Plug and Play [UPnP]. The ZAL (and CAP) differs from these other standards because it was designed to have a compact message format and a simple implementation. By comparison, these two related standards use HTTP and XML as a data exchange format. HTTP/XML is more verbose than the binary messages over UDP used by CAP, which may not be suitable for low-bandwidth low-power networks. HTTP/XML requires more host resources to encode, decode, and process, which may not be suitable for extremely resource-constrained IP-connected hosts running on microcontrollers.

1.1. Usage Model

An example of a scenario to which CAP may be applied is a "smart energy" system -- a connected set of energy consuming devices, energy producing or transmitting devices, and human-interactive displays and controllers.

One such smart energy system intended for home use may include a number of load controlled appliances, a connected electricity and gas meter plus one or more submeters, an interactive wall-mounted display, and a host whose purpose is to apply an energy usage policy to the devices in the system, sometimes called an "Energy Services..."
Agreement among vendors can result in the production of an Application Profile document for the "Smart Energy" application domain, which includes a set of device classes that fit this domain, and a concrete description of the functionality provided by each device.

The devices implementing this profile document will use the CAP Core Protocol to communicate over the network and will use the CAP Data Protocol to exchange data. These protocols will be employed when the meters send energy usage information to the display, and when the energy services portal sends policy control messages to the load control units, for example. The Security Protocol portions of CAP may be used to authenticate, encrypt, and decrypt these Core Protocol messages.

When the network is being created or when new devices are being added, the network administrator will use the CAP Management Protocol to discover the existence of devices on the network and the capabilities of those devices. Then, the administrator will establish communication links between devices by manipulating a binding table resident on each device, also with the Management Protocol. This binding table contains IP addresses and ports of peer devices running CAP, and Cluster identifiers used to connect compatible interfaces.

1.2. Terms Used

ZigBee Specification The original document that describes the ZigBee Network Layer and ZigBee Application Layer [ZigBee].

ZAL (ZigBee Application Layer) An application-layer protocol and profile system for service discovery, binding, security, and structured communication. Originally designed to run over the ZigBee Network Layer. Defined in [ZigBee].

ZNL (ZigBee Network Layer) A link-level mesh protocol designed to run on the IEEE 802.15.4 link, specified in terms of IEEE 802.15.4 link layer addresses. Defined in [ZigBee].

IEEE 802.15.4 A low-power low-bandwidth link layer, over which ZigBee was originally intended to run.

ZigBee Network Address An IEEE 802.15.4 link address that is used within the ZNL to identify a host.
ZCL (ZigBee Cluster Library) A protocol that runs above the ZAL, providing a set of abstractions for remote data access through Attributes and Commands. Also, a public list of well-defined interfaces to named units of functionality that can be provided by CAP Data Peers, composed of Attributes and Commands. Defined in [ZigBeeCL].

CAP (Compact Application Protocol) This document’s name for the adapted version of the ZigBee Application Layer that runs over UDP/IP.

APS (ZigBee Application Support Layer) The lowest layer of the ZAL, defined in [ZigBee].

Core Protocol The lowest level of CAP, that corresponds to the ZigBee Application Support Layer.

CAP Address Record A structure that contains an IPv4 address and port, IPv6 address and port, or DNS hostname and port, which is used to replace the ZigBee addressing used in the original ZAL.

Data Peer The basic service role that a CAP host may play, exchanging APS messages, possibly fragmented or secured, and containing ZCL operations for Command and Attribute transmission and reception.

Data Protocol The layer of CAP that corresponds to the ZCL protocol, and runs above the Core Protocol layer.

Discovery An interaction that allows a CAP Data Peer to send its Device types and Cluster lists over the network, and allows an Administrator to look up this information in a Discovery Cache.

Discovery Cache A CAP host that provides storage and querying for the service information provided by CAP Data Peers by responding to the Discovery Protocol Messages described in the Management Protocol section of this document.

Binding A connection between two CAP Data Peers that share a common Cluster, stored in a Binding Table on the Data Peer that needs to contact the other Data Peer.

ZDP (ZigBee Device Profile) The subset of the ZAL that performs management operations like discovery and binding. Defined in [ZigBee].
Management Protocol  The layer of CAP that corresponds to the ZigBee Device Profile, providing a protocol for discovery and binding.

Trust Center  A CAP host that permits joining and leaving of a Security Domain by distributing a shared Domain Key, and acts as a key-escrow center to assign shared Link Keys to pairs of nodes that need to communicate securely.

Security Protocol  The layer of CAP that corresponds to the APS Layer Security, including a header that may be inserted into an APS message and a set of messages for key exchange.

Administrator  A CAP host that initiates binding and discovery operations in order to establish communication relationships among Data Peers.

Binding Coordinator  A CAP host that assists nodes in establishing relationships without an Administrator’s involvement, by responding to the End Device Bind messages described in the Management Protocol section.

Binding Table Cache  A CAP host that holds a copy of the binding information normally resident in the individual nodes’ binding tables, which is obtained by responding to the Binding Table Cache messages described in the Management Protocol section.

Application Profile  A document that specifies an application domain consisting of a set of Devices that are intended to work together, and the well-defined Cluster interfaces needed by these devices to communicate.

Device  A named object that implements a set of Cluster interfaces to provide app-layer functionality.

Cluster  A specific remote interface that provides a single unit of functionality composed of Attributes and Commands.

Attribute  A definition of a single data value that may be produced or consumed by a device implementing a Cluster.

Command  A definition of a single message that may be sent between devices implementing a Cluster intended to initiate an operation.

Datatype  A definition of a concrete representation of a value contained in an Attribute or Command.
Endpoint  An identifier used to enable multiple Devices to be implemented by a single CAP Data Peer.

1.3. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.4. ZigBee Notice

In order to represent a product as ZigBee, ZigBee Compliant, ZigBee Certified or similar or to use any ZigBee Alliance intellectual property (including, without limitation, the ZigBee Specification, ZigBee Profile Specifications, ZigBee Test Plans, ZigBee trademarks or logos) for any commercial purpose, proper licensing must be obtained by joining the ZigBee Alliance or entering into license agreements with the relevant members of the ZigBee Alliance.

2. Transport

As specified, the ZigBee Application Layer messages are transmitted by the ZigBee Network Layer. The first basic modification of the ZigBee Application Layer is to embed its messages in UDP datagrams instead of ZigBee Network Layer frames. Thus, CAP messages are transmitted as datagrams over UDP/IP. Note that the ZigBee Application Layer messages, and thus the CAP messages, are specified to be little-endian.

A CAP peer will typically need to be able to receive unsolicited notifications, and to do so, the CAP peer MUST bind to the chosen UDP port for listening. If the chosen port is unavailable on the host, then an alternate port MAY be chosen and used when communicating, binding, and registering. In the common case of a CAP client-server system, the chosen listening port SHOULD also be used as the source port for transmissions. Transmissions from client-only systems MAY be sent from an ephemeral port.

3. Bootstrapping

Because CAP devices often do not contain human interfaces, other mechanisms are employed to provide the IP addresses and ports of CAP nodes in the network to other CAP nodes. CAP provides a Discovery Cache, which other nodes can use to look up the addresses of CAP nodes, but the IP address and port of the Discovery Cache is typically obtained through means outside of CAP. In addition, the IP
address of the Trust Center, Binding Coordinator, and Binding Table Cache are also discovered by the CAP Data Peers typically through means outside of CAP.

Preconfiguration When the IP addresses or DNS names of the CAP servers are well-known, they can be configured into the CAP Data Peers.

DHCP Option If a DHCP server is available, it can provide the addresses of these servers via a DHCP option to be specified at some future time, as is done today to assign a DNS server address [RFC2131].

DNS-Based Service Discovery If a DNS server is available, the CAP Data Peer can look up a SRV record containing a type for the CAP protocol to be specified at some future time, with an associated set of keys in a TXT record that describe the CAP server functionality available on the host [I-D.cheshire-dnsext-dns-sd]. This service discovery operation can run via standard unicast DNS in the wide area case, or multicast DNS in the local network case [I-D.cheshire-dnsext-multicastdns].

CAP Server Discovery If these mechanisms are not available, the CAP Data Peer can send the CAP Server Discovery message via IPv4 subnet broadcast or IPv6 link-local multicast [RFC2373] to discover any CAP Servers available on the local network.

4. Address Replacement

As specified, the ZigBee Application Layer protocol identifies nodes in the network either by their globally-unique 64-bit IEEE 802.15.4 EUI-64 or by their locally-unique 16-bit IEEE 802.15.4 short address. The second basic modification applied to the ZigBee Application Layer is to replace each use of an 802.15.4-specific address with a CAP Address Record, which can contain an IPv4 address plus UDP port, IPv6 address plus UDP port, or DNS Fully-Qualified Domain Name [RFC1034] plus UDP port.

The CAP Address Record has five possible formats, which are distinguished by the value of the leading Type byte:

Source Address (0x01) A message sender includes this address record type as a placeholder in a message to indicate that the intended address is its own source IP address and source port. When this record type is encountered in a received network message, the processing node MUST use the source IP address and source port contained in the message for any further operations that include
the address. Examples of such operations include registration of
discovery information or addition of binding table entries. This
address record type is only intended for use in a message, and
MUST NOT be stored in any in-memory or persistent table such as a
discovery cache table or a binding table. Instead, the actual
source IP address and source port MUST be stored in the table,
using one of the other non-placeholder address types.

+------------------+
| Type = 0x01      |
+------------------+

Destination Address (0x02) A message sender includes this address
record type as a placeholder in a message to indicate that the
actual address is the destination IP address and destination port
of the message. When this record type is encountered in a
received network message, the processing node MUST use the
destination IP address and destination port contained in the
message for any further operations that include the address.
Examples of such operations include query of discovery
information. This address record type is only intended for use in
a message, and MUST NOT be stored in any in-memory or persistent
table such as a discovery cache table or binding table. Instead,
the actual destination IP address and destination port MUST be
stored in the table, using one of the other non-placeholder
address record types.

+------------------+
| Type = 0x02      |
+------------------+

IPv4 Address (0x03) This address record type contains a single IPv4
address and port.

+------------------+------------------+
| Type = 0x03      | IPv4 Address      |
+------------------+------------------+
| Port             |                   |
+------------------+

IPv6 Address (0x04) This address record type contains a single IPv6
address and port.
DNS Name (0x05)  This address record type contains a single DNS hostname. It may be stored in a table, but will have to be resolved to an IP address prior to every communication with the referenced node.

The CAP Address Record is always represented in standard network byte order (big-endian) when included in a CAP message.

5. Core Protocol

The basic CAP UDP message contains the ZigBee Application Layer APS frame with the format defined in Section 2.2.5 "Frame Formats" of the ZigBee Specification, and all nested payloads and additional headers that may be placed within the APS frame. This embedded frame is termed the CAP Core Protocol message.

The basic CAP interaction over UDP:

```
Data Peer                        Data Peer
on IP Address X : Port Y        on IP Address X’ : Port Y’
                                      | CAP APS Frame in UDP |
                                      | (optional) CAP APS Ack in UDP |
```

Tolle                          Expires April 11, 2009
5.1. Protocol Behavior

5.1.1. Transmission

CAP Core Protocol messages are transmitted from one IP host to another IP host. All three ZigBee APS Frame Types are allowed (APS Data, APS Command, APS Acknowledgment), and transmission proceeds according to the rules in section 2.2.5.1 of the ZigBee Specification, except as described below.

The Delivery Mode sub-field of the APS header is typically set to 0x00 (Normal Unicast Delivery), unless the destination IP address is a multicast address, in which case this field is set to 0x10 (Broadcast). Indirect delivery assumes a ZigBee Network Layer, and thus this option is not used in CAP. Because Group delivery interacts with the ZigBee Network Layer via 16-bit group addresses, CAP uses the ZigBee mode of operation described in Section 2.2.8.4.1 of the ZigBee Specification when nwkUseMulticast is true. Thus, Group delivery is reduced to Broadcast delivery with a multicast destination IP address and a destination CAP endpoint of 0xFF.

The upper-layer sender of the message specifies whether or not security should be used and with which type of key, and whether or not reliable transfer with acknowledgments should be used.

If secure transmission is requested, then the device shall follow the security processing steps specified in Section 4.4.1.1 of the ZigBee Specification, as modified by the Security Protocol section in this document, to generate a new message containing security headers.

If reliable transfer is requested, then the device shall set the Acknowledgment Request bit in the APS header, send the message, and wait up to ACK_WAIT_DURATION seconds for the acknowledgment message. If the device does not receive an acknowledgment in that time with a matching APS Counter value, it will retransmit the message (with the same APS Counter value as before) up to ACK_MAX_RETRIES times. If all transmission attempts fail, the device shall stop attempting transmission and signal the error to the upper layer.

5.1.2. Reception

The basic CAP message is received by a UDP/IP host.

If the message is secured, security processing then performed as specified in Section 4.4.1.2 of the ZigBee Specification, modified by the Security Protocol section in this document. If security processing fails, then the message MUST be discarded.
The source IP address, source port, and APS Counter field MUST be used to determine whether this message is a duplicate of a previously received message. If the message is determined to be a duplicate, then the message MUST be discarded.

If an acknowledgment has been requested by setting the Acknowledgment Request sub-field, the CAP implementation must generate the appropriate APS acknowledgment frame according to the rules in Section 2.2.5.2.3.1 of the ZigBee Specification, and send it to the source port, and source IP address of the received CAP message.

The message can now be processed by the next upper layer. If the message is an APS Command message, then it will be processed by the Security layer. If the message is a Data message, then the destination endpoint will be used to decide which layer processes the message. If the destination endpoint is 0, the CAP Management Protocol layer will process it. Otherwise, the CAP Data Protocol layer will process it. The implementations of all upper layers must have access to the values contained in all the fields of this message, in particular the source and destination endpoint identifiers, cluster identifier, and profile identifier.

6. Data Protocol

The CAP Data Protocol is contained within the APS Payload section of the CAP Core Protocol message. The Data Protocol message contains the ZCL Command Frame, as defined in section 2.3 of the ZigBee Cluster Library Specification. Hosts communicating with the CAP Data Protocol are termed Data Peers.

All ZCL Commands (e.g. Read, Write, Report, Configure Reporting, Discover Attributes) defined in Section 2.4 of the ZigBee Cluster Library Specification are valid for inclusion in the ZCL Payload section of the Data Protocol frame.

The behavior upon transmission and reception of these Data Protocol frames must follow the rules in Section 2.3 and Section 2.4 of the ZigBee Cluster Library Specification.

Because none of the ZCL Commands contain addresses, and only refer to data objects, none of the message formats require modification to run within the CAP context.
Example of Data Protocol interactions over UDP/IP

<table>
<thead>
<tr>
<th>Data Peer</th>
<th>Data Peer</th>
<th>Data Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ZCL Read Attributes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(optional) CAP APS Ack in UDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZCL Read Attributes Response</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(optional) CAP APS Ack in UDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZCL Report Attributes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(optional) APS Ack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------------------&gt;</td>
<td></td>
</tr>
</tbody>
</table>

7. Management Protocol

The CAP Management Protocol is contained within the APS Payload section of the CAP Core Protocol message. The Management Protocol message contains the ZigBee Device Profile (ZDP) Command Frame, as defined in Section 2.4.2.8 of the ZigBee Specification.

The Management Protocol contains the ZigBee Device Profile Client Services and Server Services messages defined in Section 2.4.3 and Section 2.4.4 of the ZigBee Specification, with certain exceptions described below. Exceptions are either:

- exclusion of specific messages that define operations that apply only to the original IEEE 802.15.4-based network layer and link layer, and cannot be modified to support IP networks
- modifications of specific messages to replace IEEE 802.15.4-based addressing with IP addressing
7.1. Discovery Messages

Example of Management Protocol for Registration of Discovery Information with Discovery Cache

Discovery Cache                          Joining Data Peer            Administrator

<table>
<thead>
<tr>
<th>Discovery_store_req</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery_store_rsp</td>
</tr>
<tr>
<td>(store all data with store reqs)</td>
</tr>
</tbody>
</table>

Example of Management Protocol for Request of Discovery Information from Discovery Cache

Discovery Cache                          Data Peer            Administrator

<table>
<thead>
<tr>
<th>Node_Desc_req( Data Peer )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node_Desc_rsp( Data Peer )</td>
</tr>
</tbody>
</table>

Example of Management Protocol for Request of Discovery Information from the Node.

Discovery Cache                          Data Peer            Administrator

<table>
<thead>
<tr>
<th>Node_Desc_req()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node_Desc_rsp()</td>
</tr>
</tbody>
</table>
NWK_addr_req/rsp (ClusterID=0x0000/0x8000)  This message pair is not used in CAP, because it only maps between the two types of IEEE 802.15.4 addresses, and is not needed when both peers are only using IP addresses.

IEEE_addr_req/rsp (ClusterID=0x0001/0x8001)  This message pair is also not used in CAP, as it is the inverse of NWK_addr_req.

Node_Desc_req/rsp (ClusterID=0x0002/0x8002)  The request and response each contain a single ZigBee Network Address in a field named NWKAddrOfInterest, which is the address of the node whose Node Descriptor is being requested. For use in CAP, replace the NWKAddrOfInterest with a single CAP Address Record.

Power_Desc_req/rsp (ClusterID=0x0003/0x8003)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Simple_Desc_req/rsp (ClusterID=0x0004/0x8004)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Active_EP_req/rsp (ClusterID=0x0005/0x8005)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Match_Desc_req/rsp (ClusterID=0x0006/0x8006)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Complex_Desc_req/rsp (ClusterID=0x0010/0x8010)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

User_Desc_req/rsp (ClusterID=0x0011/0x8011)  Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Discovery_Cache_req (ClusterID=0x0012/0x8012)  This request message is sent to a node or to a multicast address to determine whether the node supports the Discovery Cache service. In the ZigBee Specification, the message contains the two source ZigBee Network Addresses, to which the Discovery_Cache_rsp will be sent. In CAP, replace these two addresses with a single CAP Address Record representing the sender of the message. The response can be used as-is, because it does not contain any addresses.
Device_annce (ClusterID=0x0013) This message is not used in CAP, because it relates solely to ZigBee Network Layer address assignment.

User_Desc_set/conf (ClusterID=0x0014/0x8014) Replace NWKAddrOfInterest with CAP Address Record in both request and response.

System_Server_Discovery_req/rsp (ClusterID=0x0015) This message pair does not contain any addresses, and so it is used as-is. However, this message should only be used as a fallback for multicast server address discovery in CAP. Typically, server address discovery should be performed using other out-of-band IP mechanisms, including but not limited to pre-assignment of the address, DHCP delivery of the address, or DNS Service Discovery to obtain the address.

Discovery_store_req/rsp (ClusterID=0x0016/0x8016) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. The response can be used as-is.

Node_Desc_store_req/rsp (ClusterID=0x0017/0x8017) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. The response can be used as-is.

Power_Desc_store_req/rsp (ClusterID=0x0018/0x8018) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. Remove the IEEEAddr from the response, because it is not needed.

Active_EP_store_req/rsp (ClusterID=0x0019/0x8019) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. The response can be used as-is.

Simple_Desc_store_req/rsp (ClusterID=0x001a/0x801a) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. The response can be used as-is.

Remove_node_cache_req/rsp (ClusterID=0x001b/0x801b) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. The response can be used as-is.

Find_node_cache_req/rsp (ClusterID=0x001c/0x801c) Replace the two Local Device ZigBee Network Addresses with a single CAP Address Record in the request. In the response, replace CacheNWKAddress with a CAP Address Record representing the address of the Discovery Cache holding the information. Replace the two "device of interest" addresses with a single CAP Address Record.
representing the device being queried for.

Extended_Simple_Desc_req/rsp (ClusterID=0x001d/0x801d) Replace NWKAddrOfInterest with CAP Address Record in both request and response.

Extended_Active_EP_req/rsp (ClusterID=0x001e/0x801e) Replace NWKAddrOfInterest with CAP Address Record in both request and response.

### 7.2. Binding Messages

Example of Management Protocol for Administratively Binding Two Data Peers Together

<table>
<thead>
<tr>
<th>Data Peer X</th>
<th>Data Administrator</th>
<th>Data Peer Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bind_req( Cluster, Y )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;===================================</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bind_rsp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>===================================</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZCL Report Attributes( Cluster )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>===================================</td>
<td></td>
</tr>
</tbody>
</table>
Example of Management Protocol for End Device Binding Assisted by the Binding Coordinator

End_Device_Bind_req (ClusterID=0x0020/0x8020) In the request, replace the BindingTarget field with a CAP Address Record representing the address of the node that will hold the binding table entry (typically the node’s own address, unless a binding table cache is in use). Replace the SrcIEEEAddress field with another CAP Address Record representing the source address of the binding, which is the node’s own address. The response just contains a Status field, so it may be used as-is.

Bind_req (ClusterID=0x0021/0x8021) In the request, replace the SrcAddress field with a CAP Address Record containing the source address of the binding. Replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the
destination address of the binding. The DstEndp field must be present. The response just contains a Status field, so it may be used as-is.

Unbind_req (ClusterID=0x0022) In the request, replace the SrcAddress field with a CAP Address Record containing the source address of the binding. Replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present. The response just contains a Status field, so it may be used as-is.

7.3. Binding Table Cache Messages

Bind_Register_req/rsp (ClusterID=0x0023/0x8023) In the request, replace the NodeAddress field with a single CAP Address Record. In each element of the BindingTableList array field in the response, replace the SrcAddress field with a CAP Address Record containing the source address of the binding, and replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present in each entry.

Replace_Device_req/rsp (ClusterID=0x0024/0x8024) In the request, replace the OldAddress field with a single CAP Address Record. Replace the NewAddress field with a single CAP Address Record. The response just contains a Status field, so it may be used as-is.

Store_Bkup_Bind_Entry_req/rsp (ClusterID=0x0025/0x8025) In the request, replace the SrcAddress field with a CAP Address Record containing the source address of the binding. Replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present. The response just contains a Status field, so it may be used as-is.

Remove_Bkup_Bind_Entry_req/rsp (ClusterID=0x0026/0x8026) In the request, replace the SrcAddress field with a CAP Address Record containing the source address of the binding. Replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present. The response just contains a Status field, so it may be used as-is.

Backup_Bind_Table_req/rsp (ClusterID=0x0027/0x8027) In each element of the BindingTableList array field in the request, replace the SrcAddress field with a CAP Address Record containing the source address of the binding, and replace the DstAddrMode field and
The DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present in each entry. The response may be used as-is.

Recover_Bind_Table_req/rsp (ClusterID=0x0028/0x8028) This request can be used as-is, because it does not include any addresses. In each element of the BindingTableList array field in the response, replace the SrcAddress field with a CAP Address Record containing the source address of the binding, and replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present in each entry.

Backup_Source_Bind_req/rsp (ClusterID=0x0029/0x8029) In the request, replace each element of the SourceTableList with a CAP Address Record. The response just contains a Status field, so it may be used as-is.

Recover_Source_Bind_req/rsp (ClusterID=0x002a/0x802a) This request can be used as-is, because it does not include any addresses. In the response, replace each element of the SourceTableList with a CAP Address Record.

7.4. Network Management Messages

Mgmt_NWK_Disc_req/rsp (ClusterID=0x0030/0x8030) This message pair is not used in CAP, because it only relates to the ZigBee Network Layer.

Mgmt_Lqi_req/rsp (ClusterID=0x0031/0x8031) This message pair is not used in CAP, because it is just a request for a table of IEEE 802.15.4 link quality measurements.

Mgmt_Rtg_req/rsp (ClusterID=0x0032/0x8032) This message pair is not used in CAP, because it is just a request for a table of ZigBee Network Routes.

Mgmt_Bind_req/rsp (ClusterID=0x0033/0x8033) This request is used as-is. In each element of the BindingTableList array field in the response, replace the SrcAddress field with a CAP Address Record containing the source address of the binding, and replace the DstAddrMode field and DstAddress field with a single CAP Address Record representing the destination address of the binding. The DstEndp field must be present in each entry.
Mgmt_Leave_req/rsp (ClusterID=0x0034/0x8034) This message pair is not used in CAP, because it is a request to leave a ZigBee Network.

Mgmt_Direct_Join_req/rsp (ClusterID=0x0035/0x8035) This message pair is not used in CAP, because it is a request to join a ZigBee Network.

Mgmt_Permit_Joining_req/rsp (ClusterID=0x0036/0x8036) This message pair is not used in CAP, because it is a request to enable another node to permit nodes to join a ZigBee network.

Mgmt_Cache_req/rsp (ClusterID=0x0037/0x8037) This request is used as-is. In each element of the DiscoveryCacheList array field in the response, replace the two ZigBee Addresses with a single CAP Address Record.

Mgmt_NWK_Update_req/rsp (ClusterID=0x0038/0x8038) This message pair is not used in CAP, because it only relates to the ZigBee Network Layer.

8. Security Protocol

The CAP Security Protocol is based on the ZigBee APS Layer Security defined in Section 4.4 of the ZigBee Specification.

Because all dependencies on the ZigBee Network Layer have removed in CAP, the adapted Security Protocol is used only between nodes that can already reach each other over the IP network layer. In general, ZigBee Nodes rely on their Network Layer parent to authenticate with the Trust Center on their behalf. In CAP, this model is not used. A node may initiate secure communications or authenticate itself to the Trust Center at any time after joining the IP Network.

8.1. Secure Communication

The basic element of the Security Protocol is a Security Header that may be included in the Core Protocol message, which corresponds to the Auxiliary Frame Header defined in Section 4.5.1 of the ZigBee Specification. This Security Header is included after the Core Protocol header, and a Security Message Authentication Code is included after the payload of the message. These headers contain authentication and encryption information needed to verify or decrypt the payload. Authenticated and encrypted communication may occur between any two CAP nodes using shared keys. In CAP, the Extended Nonce bit MUST be set to 1, because a Source Address is always included in the nonce.
Secure communication may proceed between any pair of CAP nodes, as long as they share one of the three ZigBee key types defined in Section 4.2.1.3 of the ZigBee Specification: a Link Key, a Master Key, or a Network Key. All three key types are 128-bits in size. The Link Key and Master Key are only shared to a single pair of nodes, and in CAP, the endpoints are represented by CAP Address Records. The Network Key is defined by the ZigBee Standard as a key shared between all nodes within the IEEE 802.15.4 network, and available for use by both the ZigBee Network Layer and the ZigBee Application Layer. In the absence of a ZigBee Network Layer, the Network Key is redefined here as a Domain Key that is shared between a set of CAP devices that comprise a common Security Domain, defined solely as the set of CAP devices communicating with a particular Trust Center host and sharing a common Domain Key.

The rules for generating the Security Header when performing security processing on outgoing messages is defined in Section 4.4.1.1 of the Zigbee Specification. Because the Domain Key does not depend on any addresses, it may be used exactly as defined to secure both unicast and multicast transmissions. Link Keys and Master Keys shared between the message sender and particular devices, and are stored by CAP in a table called the "apsDeviceKeyPairSet", defined in Section 4.4.10 of the ZigBee Specification. For CAP purposes, each key-pair descriptor shall have the following fields instead of those defined in Table 4.37:

DeviceAddress  is the CAP Address Record containing the remote address of the CAP peer.

MasterKey  is the 16-byte value of the master key shared with the peer at DeviceAddress

LinkKey  is the 16-byte value of the link key shared with the peer at DeviceAddress

OutgoingFrameCounter  is an unsigned 32-bit counter for use with this key

IncomingFrameCounter  is an unsigned 32-bit counter corresponding to the DeviceAddress

In Step 2 of the Outgoing Frame processing steps, the APS layer shall always apply security when security is requested, because ZigBee Network Layer security does not exist in CAP.

In Step 4, the CAP implementation shall internally maintain the Security Level that is in use among the devices within the Security Domain.
The Security Auxiliary Header shall be constructed according to the rules in Steps 5 through 11. The CAP implementation must maintain an 8-byte Source Identifier, and use it when constructing the nonce \( N \) as defined in Section 4.5.2.2. This source identifier MUST be derived from an available link-layer identifier, like an EUI-64, or a MAC-48 that has been expanded into an EUI-64 according to the rules defined in [RFC5342]. This same Source Identifier must then be included in the Source Address field of the Security Auxiliary Header, and the Extended Nonce subfield must be set to 1.

8.2. Key Management Protocol

The ZigBee Application Layer includes a Trust Center, which is preserved in CAP. The Trust Center is responsible for:

- accepting authentication requests from CAP nodes and maintaining a table of authenticated nodes
- generating and updating a shared Domain Key for a set of CAP nodes
- accepting requests to generate and securely deliver shared link keys to individual pairs of CAP nodes that wish to communicate securely

To avoid transmitting key material in the clear, a node that is intended to join a Security Domain MUST be preconfigured with one or more shared keys relevant to the Security Domain.

In Standard Security Mode, the node MAY be preconfigured with the Domain Key, which allows the node to communicate securely with other CAP nodes sharing the same key, but does not provide any per-node authentication. Or, the node MAY be preconfigured with a Link Key shared between itself and the Trust Center. This key is used to securely join the domain and request the Domain Key. As long as the node has one of these keys, it can make use of the Trust Center to establish node-to-node Link Keys in order to secure particular pairwise communications.

In High Security Mode, the node MAY be preconfigured with a Master Key shared between itself and the Trust Center. This Master Key is then used to mutually establish a Link Key according to the SKKE mechanisms described in Section 4.4.2.6 of the ZigBee Specification. The Trust Center MAY then generate and deliver Master Keys to pairs of CAP nodes, and those nodes may also make use of SKKE to mutually establish Link Keys.

To support a scenario in which pre-shared keys are not feasible, other key exchange mechanisms outside of CAP MAY be employed to
securely agree on shared keys over the network. The shared keys may then be employed as ZigBee Link Keys or Master Keys to secure future communications.

8.2.1. Key Management Protocol Examples

This section gives examples of the proper usage of the ZigBee security services in the end-to-end UDP/IP context used by CAP nodes. In general, the behaviors defined in Section 4.6 are used, but modified so that the Data Peer is directly interacting with the Trust Center, instead of relying on its ZigBee Network Layer parent to perform the authentication. For protocol exchanges that do not rely on the ZigBee Network Layer, such as Update_Key, then no modification is required.

Join Domain with Preconfigured Domain Key

<table>
<thead>
<tr>
<th>Trust Center</th>
<th>Joining Data Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain_kt[ Update_Device( 00 {Standard, Secured, Rejoin} ) ]</td>
<td>&lt;===============================================</td>
</tr>
<tr>
<td>Domain_kt[ Transport_Key( 01 Std Domain, Key=0, Seq=0 ) ]</td>
<td>==============================================&gt;</td>
</tr>
</tbody>
</table>

When a new device is preconfigured with the correct Domain Key and Domain Key Sequence Number, it MAY send a message secured with that key in order to inform the Trust Center of its presence in the network.

Join Domain with Preconfigured Trust Center Link Key

<table>
<thead>
<tr>
<th>Trust Center</th>
<th>Joining Data Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC_Link_kt[ Update_Device( 00 {Standard, Sec, Rejoin} ) ]</td>
<td>&lt;===============================================</td>
</tr>
<tr>
<td>TC_Link_kt[ Transport_Key( 01 Std Domain, Key=k, Seq=s ) ]</td>
<td>==============================================&gt;</td>
</tr>
</tbody>
</table>

When a new device is preconfigured with a Link Key that has been shared with the Trust Center, it MAY send a message secured with that key.
key in order to inform the Trust Center of its presence and then receive the current Domain Key.

Request the Active Domain Key

<table>
<thead>
<tr>
<th>Trust Center</th>
<th>Joined Data Peer</th>
</tr>
</thead>
</table>
| TC_Link_kt[ Request_Key( 01 Domain ) ] <==================================>
| TC_Link_kt[ Transport_Key( 01 Std Domain, Key=k, Seq=s ) ] ==============>

A new device may request the current Domain Key at any time, using an interaction secured by a shared Link Key with the Trust Center.

Distribute a new Domain Key

<table>
<thead>
<tr>
<th>Trust Center</th>
<th>Joined Data Peer</th>
</tr>
</thead>
</table>
| kt [ Transport_Key( 01 Std Domain, Key=k’, Seq=s+1 ) ] ===============>

Store Alternate Key

| kt [ Switch_Key( Seq=s+1 ) ] ------------------------------------------>

Use New Key

The Trust Center MAY distribute a new Domain Key to each node in its registered list using unicast UDP/IP messages. If IP multicast is available, then the Trust Center MAY also distribute the new Domain Key using multicast. If delivered via unicast, the message MUST be secured with the specific Link Key shared with each receiving node. If delivered via multicast, the message MUST be secured with the shared Domain Key. This key is then held by the nodes until they...
receive a Switch Key message secured according to the rules above, at which time the new key is activated. When activating the new key, the node resets the incoming frame counters and outgoing frame counter to 0. If the node is not capable of holding two keys at the same time, then the new key MAY be activated immediately without waiting for the Switch Key message.

Distribute a new Trust Center Link or Master Key

The Trust Center MAY send a new Link Key or Master Key to a CAP node using the Transport Key message. Upon reception, the CAP node MUST replace the Trust Center’s entry in the apsDeviceKeyPairSet table.

Request Node-To-Node Application Link Keys
A node that wishes to initiate secure communication with another node sharing the same Trust Center may request that the Trust Center generate a shared Link Key for the pair of nodes. The Key $k$ that gets sent to the nodes MUST be randomly generated. When the nodes receive the key, they MUST add it to the peer node’s entry in the `apsDeviceKeyPairSet` table. These messages MUST be secured by the Trust Center’s Link Key for the receiving node. In High-Security Mode, the Trust Center MAY distribute a shared Master Key as well.

### Leaving the Domain

![Diagram of Trust Center and Leaving Data Peer with Update_Device(02 Device Left) message]

A device MAY inform the Trust Center that it is unregistering for Domain Key updates and Link Key delivery by sending an `Update Device` message with the above status code.

#### Telling a Device to Leave the Domain

![Diagram of Trust Center and Leaving Data Peer with Remove_Device(Leaving Node) message]

The Trust Center MAY also inform a device that it is no longer allowed within the Security Domain by sending it a `Remove Device` message. The device will no longer receive Domain Key updates or Link Key deliveries after this message has been sent and the node’s entry has been removed from the Trust Center’s table.

### 8.2.2. Key Management Protocol Messages

Some of the APS messages used for key management are also adapted to support the IP addresses used by CAP.

**APS_CMD_SKKE_1** (0x01) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.
APS_CMD_SKKE_2 (0x02) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.

APS_CMD_SKKE_3 (0x03) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.

APS_CMD_SKKE_4 (0x04) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.

APS_CMD_TRANSPORT_KEY (0x05) This message carries Key Descriptors. The message does not require modification, but the Key Descriptors themselves are modified because they include ZigBee EUI-64 Addresses. The modified Key Descriptors are below. Note that the discussion about the UseParent and ParentAddress operation parameters only applies to ZigBee (when a routing parent authenticates on behalf of a joining device), and so those parameters can be ignored.

Trust Center Master Key (0x00) Replace the Destination and Source Addresses with CAP Address Records.

Standard Domain Key (0x01) Replace the Destination and Source Addresses with CAP Address Records.

Application Master Key (0x02) Replace the Partner Address with a CAP Address Record.

Application Link Key (0x03) Replace the Partner Address with a CAP Address Record.

Trust Center Link Key (0x04) Replace the Destination and Source Addresses with CAP Address Records.

High-Security Domain Key (0x05) Replace the Destination and Source Addresses with CAP Address Records.

APS_CMD_UPDATEDEVICE (0x06) This message is sent to the Trust Center by a node that is joining the security domain. It includes the two ZigBee Source Addresses (Device Address and Device short address). These two fields are replaced with a single CAP address record representing the device sending the message. The Status field remains unchanged.
APS_CMD_REMOVE_DEVICE (0x07) This message is used to tell a node to leave the security domain. It contains the address of that node, which is replaced by a CAP Address Record.

APS_CMD_REQUEST_KEY (0x08) This message is used to request the active domain key from the Trust Center. The message is also used to request that a pair of application link keys be delivered to a pair of nodes in the domain, so the nodes can communicate securely. When the key being requested is an Application Link Key and the Partner Address is included, the Partner Address is replaced with a CAP Address Record.

APS_CMD_SWITCH_KEY (0x09) This message tells a node to start using a particular domain key. It contains no addresses, and so it may be used as-is.

APS_CMD_EA_INIT_CHLNG (0x0a) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.

APS_CMD_EA_RSP_CHLNG (0x0b) This message includes an Initiator Address and a Responder Address. Each one is replaced with a CAP Address Record representing the appropriate node.

APS_CMD_EA_INIT_MAC_DATA (0x0c) This message does not contain any addresses, and is used without modification.

APS_CMD_EA_RSP_MAC_DATA (0x0d) This message does not contain any addresses, and is used without modification.

APS_CMD_TUNNEL (0x0e) This message contains a Destination Address, which is replaced by a CAP Address Record.

9. Security Considerations

CAP relies on UDP/IP for transport between nodes, which is insecure by default. To remedy this, CAP adapts the ZigBee Application Layer Security system to provide authentication and encryption for CAP traffic between nodes.

If network-layer security is available, such as IPsec [RFC2411], CAP peers may make use of it to protect their messages.

As written, the APS Layer Security system relies on shared keys, which may become unwieldy in large deployments. For this reason, alternate key-agreement mechanisms may be used to establish shared keys which are then used by CAP.
As written, the APS Layer Security system relies on the presence of a Trust Center to store and manage keys on behalf of the nodes in the network. This Trust Center represents a single point of possible attack, and if this risk is unacceptable to the application designer, alternate key management mechanisms should be considered.

10. IANA Considerations

CAP requires a UDP port number to operate. An application will be submitted for allocation of a well-known UDP port number to CAP.

11. References

11.1. Normative References


11.2. Informative References


Author’s Address

Gilman Tolle
Arch Rock Corporation
501 2nd St. Ste 410
San Francisco  94107
US

Phone: 415-692-0828
Email: gtolle@archrock.com
URI:  http://www.archrock.com