

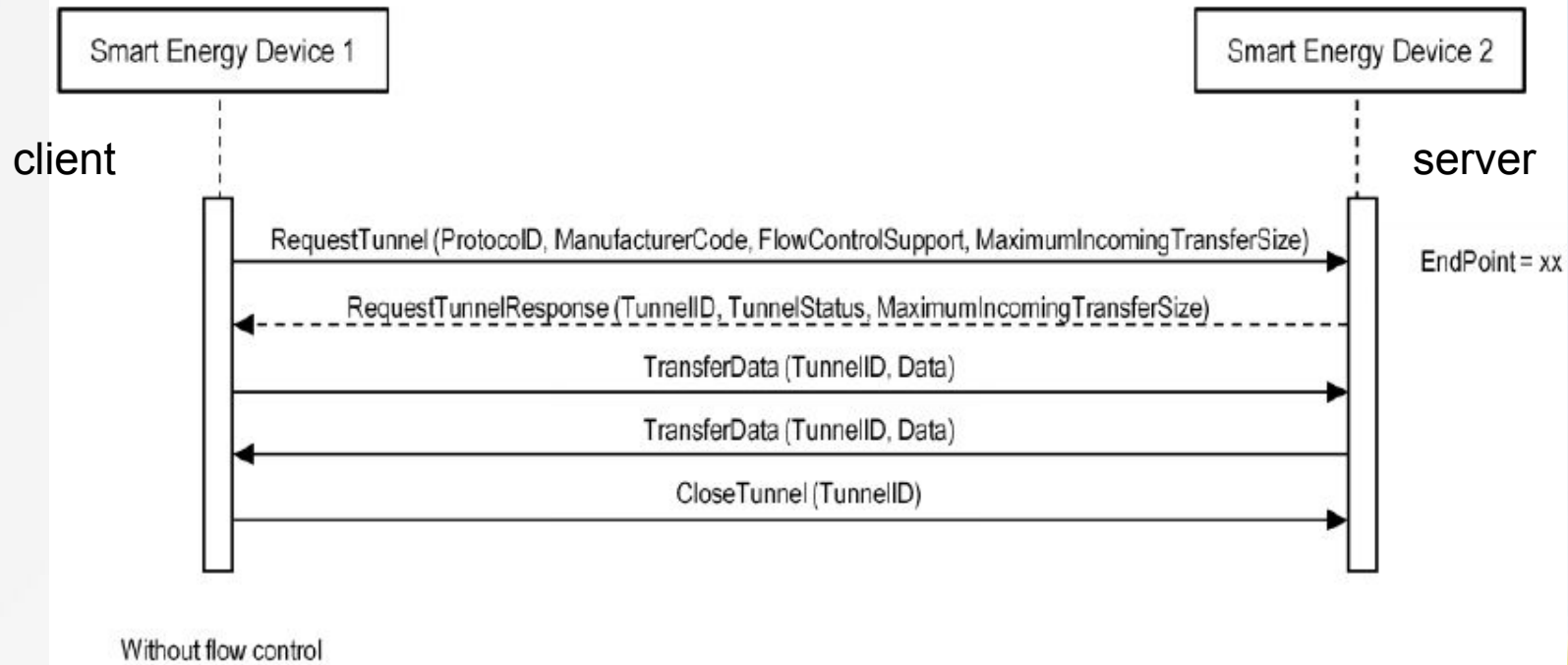
Tunneling Cluster

The background of the slide features a gradient from dark blue at the top to light blue and white at the bottom. Overlaid on this are several wavy, horizontal bands of color, including a prominent yellow band and a light blue band, creating a sense of depth and movement.

Overview:

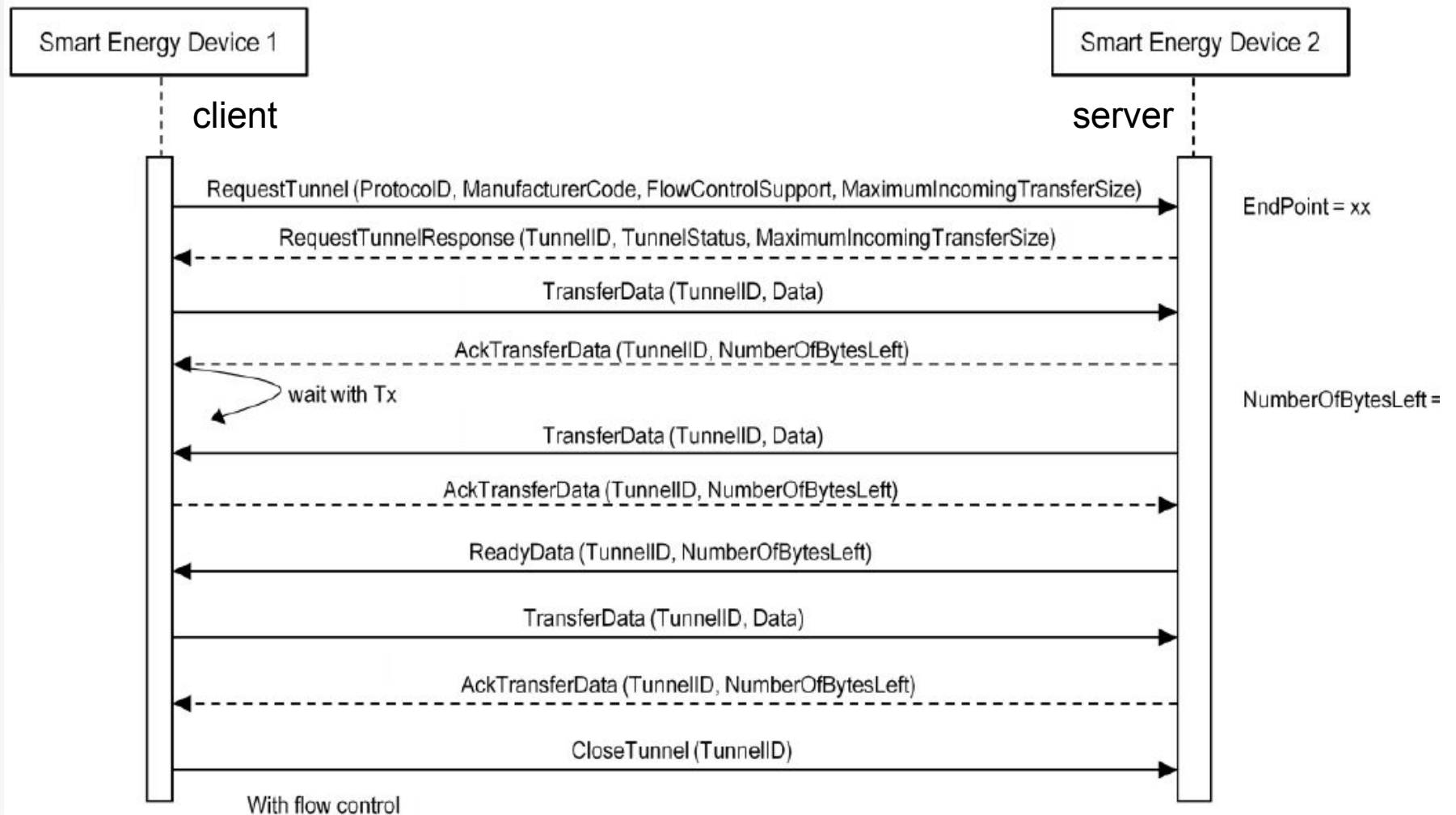
- The tunneling cluster provides an interface for **tunneling protocols**.
- **Tunneling protocols**: Tunneling enables the encapsulation of a packet from one type of protocol within the datagram of a different protocol.
- **Client**: Requests a tunnel from the server and closes the tunnel if it is no longer needed.
- **Server**: Provides and manages tunnels to the clients.
- The server may support more than one tunneling protocol.





SE **Device 1** (Client) Requests a Tunnel From SE **Device 2** (Server) to Transfer Data Without Flow Control

1. The client issues a service discovery to find devices which support the tunneling server cluster.
2. The response to the discovery from the server contains an endpoint number. Using this endpoint, the client directs a tunnel request to a given server.
3. If the server supports the protocol, it allocates the required resources, assigns a tunnel identifier and returns the ID number.
4. Both server and client may exchange data. In case the optional flow control is utilized, each data transfer is acknowledged.
5. After the transfer has been successfully completed, the client closes the tunnel again freeing the tunnel identifier in the server. If not, the server closes the 9959 tunnel by itself after CloseTunnelTimeout seconds.



SE **Device 1** (Client) Requests a Tunnel From SE **Device 2** (Server) to Transfer Data With Flow Control

Attributes:

Identifier	Name	Type	Range	Access	Default	Man. /Opt.
0x0000	<i>CloseTunnelTimeout</i>	Unsigned 16-bit Integer	0x0001- 0xFFFF	Read Only	0xFFFF	M

CloseTunnelTimeout defines the minimum number of seconds that the server waits on an inactive tunnel before closing it on its own and freeing its resources (without waiting for the CloseTunnel command from the client). Inactive means here that the timer is re-started with each new reception of a command. **0x0000 is an invalid value.**

Parameters: {not attributes}

Summary of all parameters passed to or returned by the server commands.

Name	Type	Range	Default	Mandatory / Optional
ProtocolID	8-bit enumeration	0x01 – 0xFF	0x00	M
ManufacturerCode	Unsigned 16-bit integer	0x0000 – 0xFFFF	0x00	M
FlowControlSupport	Boolean	TRUE or FALSE	FALSE	M
MaximumIncomingTransferSize	Unsigned 16-bit integer	0x0000 – 0xFFFF	1500	M
TunnelID	Unsigned 16-bit integer	0x0000 – 0xFFFF	(Return value)	M
Data	Octet string	-	-	M
NumberOfOctetsLeft	Unsigned 16-bit integer	0x0000 – 0xFFFF	-	M
TunnelStatus	Unsigned 8-bit integer	0x00 – 0x04	-	M
TransferDataStatus	Unsigned 8-bit integer	0x00 – 0x01	-	M

Note: This parameters cannot be read or written via ZCL global commands.

Commands Received: {received by the server}

Command Identifier FieldValue	Description	Mandatory/ Optional
0x00	<i>RequestTunnel</i>	M
0x01	<i>CloseTunnel</i>	M
0x02	<i>TransferData</i>	M
0x03	<i>TransferDataError</i>	M
0x04	<i>AckTransferData</i>	O
0x05	<i>ReadyData</i>	O
0x06	<i>GetSupportedTunnelProtocols</i>	O

Commands Generated: {generated by the server}

Command Identifier FieldValue	Description	Mandatory/ Optional
0x00	<i>RequestTunnelResponse</i>	M
0x01	<i>TransferData</i>	M
0x02	<i>TransferDataError</i>	M
0x03	<i>AckTransferData</i>	O
0x04	<i>ReadyData</i>	O
0x05	<i>Supported Tunnel Protocols Response</i>	O
0x06	<i>TunnelClosureNotification</i>	O

RequestTunnel Command: Is the client command used to setup a tunnel association with the server.

Payload Format:

Octets	1	2	1	2
Data Type	8-bit enumeration	Unsigned 16-bit integer	Boolean	Unsigned 16-bit integer
Field Name	ProtocolID (M)	Manufacturer Code (M)	FlowControl Support (M)	Maximum Incoming TransferSize (M)

ProtocolID: An enumeration representing the identifier of the metering communication protocol or which the tunnel is requested. The values above 199 may be used for manufacturer specific protocols.

ManufacturerCode: A code that is allocated by the ZigBee Alliance, relating the manufacturer to a device and for the tunneling a manufacturer specific protocol.

FlowControlSupport: A boolean type parameter that indicates whether flow control support is requested from the tunnel (TRUE) or not (FALSE).

0	DLMS/COSEM (IEC 62056)
1	IEC 61107
2	ANSI C12
3	M-BUS
4	SML
5	ClimateTalk
6	GB-HRGP
7 to 199	Reserved for future growth
200 to 254	Manufacturer-defined protocols
255	Reserved

MaximumIncomingTransferSize: A value that defines the size, in octets, of the maximum data packet that can be transferred to the client in the payload of a single TransferData command.

When Generated: Is never generated by the server.

Effect on Receipt: Triggers a process within the server to allocate resources and build up a new tunnel. A *RequestTunnelResponse* is generated and sent back to the client containing the result of the *RequestTunnel* command.

■ *RequestTunnelResponse Command*:

Octets	2	1	2
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	TunnelStatus (M)	Maximum Incoming TransferSize (M)

TunnelID: A number between 0..65535 that uniquely identifies the tunnel that has been allocated in the server triggered through the *RequestTunnel* command.

TunnelStatus: The TunnelStatus parameter indicates the server's internal status after the execution of a *RequestTunnel* command.

TunnelStatus:

Value	Description	Remarks
0x00	Success	The tunnel has been opened and may now be used to transfer data in both directions.
0x01	Busy	The server is busy and cannot create a new tunnel at the moment. The client may try again after a recommended timeout of 3 minutes.
0x02	No more tunnel IDs	The server has no more resources to setup requested tunnel. Clients should close any open tunnels before retrying.
0x03	Protocol not supported	The server does not support the protocol that has been requested in the ProtocolID parameter of the <i>RequestTunnel</i> command.
0x04	Flow control not supported	Flow control has been requested by the client in the <i>RequestTunnel</i> command but cannot be provided by the server (missing resources or no support).
0x05 to 0xFF	Reserved	Should not be returned and indicates an unknown error.

MaximumIncomingTransferSize: A value that defines the size, in octets, of the maximum data packet that can be transferred to the server in the payload of a single *TransferData* command.

- CloseTunnel Command: Client command used to close the tunnel with the server.

Octets	2
Data Type	Unsigned 16-bit integer
Field Name	TunnelID (M)

Effect on Receipt: In case the given TunnelID is correct, the server closes the tunnel and frees the resources associated tunnel is no longer maintained. If the TunnelID value does not match an active tunnel on the server, the server shall return a ZCL Default Response with status NOT_FOUND.

- TransferData Command: Indicates (if received) that the client has sent data to the server.

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

Data: Octets containing the data to be transferred through the tunnel in the format of the communication protocol for which the tunnel has been requested and opened.

When Generated: whenever the server wants to tunnel protocol data to the client.

Effect on Receipt: the server has received tunneled protocol data from the client.

■ TransferDataError Command:

This command is generated by the receiver of a TransferData command if the tunnel status indicates that something is wrong.

There are three cases in which TransferDataError is sent:

1. The TransferData received contains a TunnelID that does not match to any of the active tunnels of the receiving device. This could happen if a (sleeping) device sends a TransferData command to a tunnel that has been closed by the server after the CloseTunnelTimeout.
2. The TransferData received contains a proper TunnelID of an active tunnel, but the device sending the data does not match to it.
3. The TransferData received contains more data than indicated by the MaximumIncomingTransferSize of the receiving device.

Payload Format:

Octets	2	1
Data Type	Unsigned 16-bit integer	Unsigned 8-bit integer
Field Name	TunnelID (M)	TransferDataStatus (M)

Value	Description	Remarks
0x00	No such tunnel	The <i>TransferData</i> command contains a TunnelID of a non-existent tunnel.
0x01	Wrong device	The <i>TransferData</i> command contains a TunnelID that does not match the device sending the data.
0x02	Data overflow	The <i>TransferData</i> command contains more data than indicated by the <i>MaximumIncomingTransferSize</i> of the receiving device
0x03 – 0xFF	Reserved	Should not be returned and indicates an unknown error.

TransferDataStatus Values

When Generated: Is generated if the server wants to tell the client that there was something wrong with the last TransferData command.

Effect on Receipt: Indicates that the client wants to tell the server that there was something wrong with the last TransferData command.

■ **AckTransferData Command:** Command sent in response to each TransferData command.

Payload Format:

Octets	2	2
Data Type	Unsigned 16-bit integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	NumberOfBytes Left (M)

NumberOfBytesLeft : Indicates the number of bytes that may still be received by the initiator of this command (receiver).

example: A value of 150 indicates that the next TransferData command must not contain more than 150 bytes of payload or data will get lost. A value of 0 indicates that there is no more space left in the receiver and the sender should completely stop sending data.

When Generated: If flow control is on, the command is issued by the server to inform the client that the last TransferData command has been successfully received and how much space is left to receive further data.

■ Get Supported Tunnel Protocols Command:

Provides a way to read out the supported protocols from the server.

Payload format:

Octets	1
Data Type	Unsigned 8-bit Integer
Field Name	Protocol Offset

Protocol Offset: Where there are more protocols supported than can be returned in a single Supported Tunnel Protocols Response command, this field allows an offset to be specified on subsequent Get Supported Tunnel Protocols commands.

Effect on Receipt: On receipt of this command, a device will respond with a *Supported Tunnel Protocols Response command*, indicating the tunnel protocols it supports.

■ Supported Tunnel Protocols Response Command:

The response contains a list of tunnel protocols supported by the device; the payload of the response should be capable of holding up to 16 protocols.

Octets	1	1	3	...	3
Data Type	Boolean	Unsigned 8-bit Integer			
Field Name	Protocol List Complete	Protocol Count	Protocol 1	...	Protocol n



Octets	2	1
Data Type	Unsigned 16-bit Integer	8-bit Enumeration
Field Name	Manufacturer Code	Protocol ID

.....Protocol n

Protocol List Complete: {Boolean}

0 = more supported protocols available (if more than 16 protocols are supported).

1 = the list of supported protocols is complete.

Protocol Count: The number of Protocol fields contained in the response.

■ *ReadyData Command:*

Octets	2	2
Data Type	Unsigned 16-bit integer	Unsigned 16-bit Integer
Field Name	TunnelID (M)	NumberOfOctets Left (M)

When Generated: If generated by the server, this command informs the client that it may now continue to send and how much space is left within the server to receive further data.

Effect on Receipt: If received by the server, this command informs the server that it may now continue to send and how much space is left within the client to receive further data.

■ TransferData Command:

Octets	2	Variable
Data Type	Unsigned 16-bit integer	Octets
Field Name	TunnelID (M)	Data (M)

■ TunnelClosureNotification Command:

TunnelClosureNotification is sent by the server to indicate that a tunnel has been closed due to expiration of a CloseTunnelTimeout.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	TunnelID (M)

When Generated: The command is sent by a server when a tunnel is closed due to expiration of CloseTunnelTimeout. It is sent unicast to the client that had originally requested that tunnel.

CLIENT:

Attributes: The client has no attributes.

Commands Received: The client receives the cluster-specific response commands.

Commands Generated: The client generates the cluster-specific commands as required by the application.

Thank You