



Quantify Communication Protocol Overview

White Paper

Version 1.0

OVERVIEW

The Quantify Distributed Network and Enactment (QDNE) is a lightweight distributed protocol that allows qDevices, software applications, mobile applications and third-party developers to interact with other qDevices on the Local Network. The QDNE protocol, at its core, is built around the concept of enactments. Enactments are multicast messages with several components, that trigger one or more actions on a qDevice or multiple qDevices.

- It is designed to enable the following key functions:
- Discover qDevices on the network
- Trigger enactments/experiences on one or more qDevices
- Query qDevices for information regarding their state - e.g. firmware, settings etc
- Query qDevices for the enactments/experiences in which they participate
- Perform local firmware updates to one or more qDevices

The lightweight, stateless nature of QDNE makes it fast and highly scalable; it is a key component to Quantify's highly distributed architecture.

PROTOCOL INFORMATION

Network Layer Protocol	IP
Session Layer Protocol	UDP
Data Transmission Type	Multicast / Unicast
Multicast Address	224.0.6.144
Control Port	Unicast - 49152 Multicast - 49153
Enactment Port(s)	50000 - 65535

DESIGN OVERVIEW

KEY BENEFITS

The Quantify Technology platform uses the distributed nature of QDNE to achieve the following key benefits:

- **No Single Points of Failure**
All enactments/experiences are completely distributed across all relevant qDevices, and each qDevice stores and processes enactments based on its locally stored configuration. There is no requirement for a central hub or gateway.
- **Lightweight**
Enactment messages are lightweight and require minimal network overhead.
- **Scale**
Each device stores its own local configuration, meaning only a single outbound enactment is generated, no matter if an enactment/experience affects one qDevice or 20.
- **Reliable**
All enactments are acknowledged, to ensure enactments are delivered to all relevant qDevices.
- **Multicast routing ready**
By using multicast, experiences could work across multiple routed broadcast domains in enterprise networks.
- **Cloud-enhanced, Not cloud-reliant**
The Quantify cloud platform has the complex task of generating and managing the configuration of each qDevice; however, all critical functionality runs on the Local Network through QDNE alone. Critical functions operate even if the Quantify Cloud is offline.



INPUTS AND OUTPUTS

In the Quantify architecture, all inputs and outputs are completely decoupled and can be distributed over one or more qDevices. This is an important distinction from other systems; it is the QDNE Enactment messages that tie inputs and outputs together.

Examples of Inputs can be one of the following:

- Buttons on a qDevice Touch Panel
- A qMotion sensor
- The qLiving mobile application
- A real-time clock event (time of day)
- A qBridge digital input
- Triggers from the Quantify Cloud
- Third-party applications on the Local Area Network

Examples of Outputs can be one of the following:

- qDimmer channels - dimmable and non-dimmable
- Buttons on a qDevice Touch Panel (LED State)
- qPower Power Outlets
- qBlind Blind Controller
- qBridge Dry Contact Relays

A BASIC EXAMPLE

In the most basic example, two inputs (across multiple devices) are associated with a single channel output on one qDevice. This could represent a hallway with a Touch Panel at each end, controlling a single circuit of hallway lights.

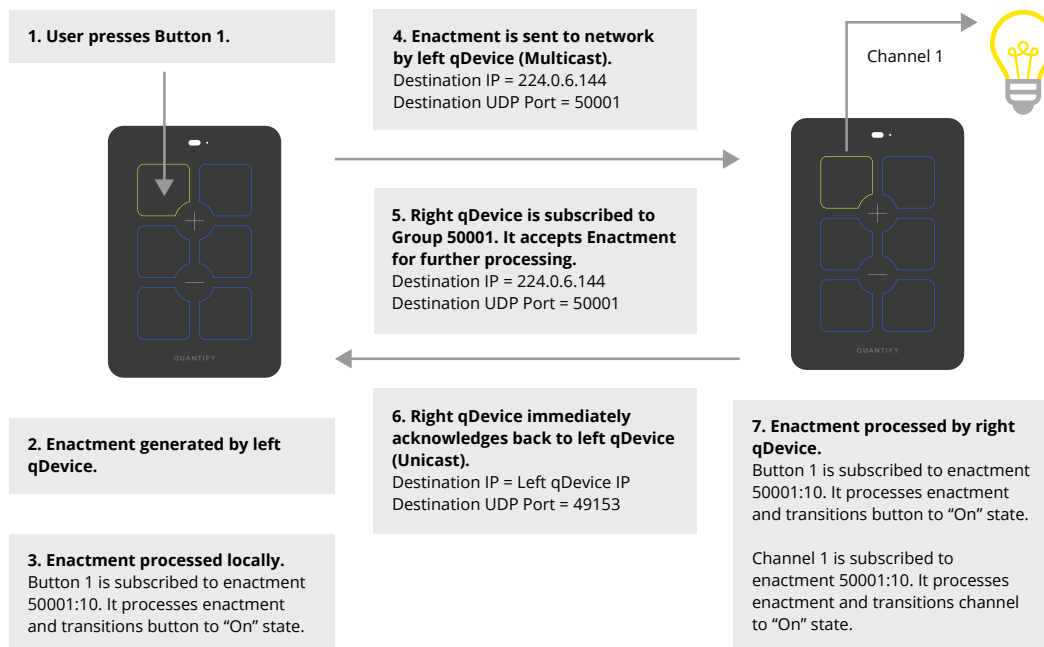


Figure 1

Steps 1-3

Touching Button 1 on the left qDevice causes an enactment to be generated – 50001:10 with a value of 100 (On). The left qDevice processes the enactment locally and transitions Button 1 to the "On" State.



Step 4

The generated enactment is also sent to the network multicast address, with the destination UDP port set to that of the Enactment Group (50001 in this case).

All qDevices on the network (even ones not pictured) will receive this message, but it will be dropped by the network stack of the qDevice if it isn't subscribed to that Enactment Group.

Steps 5 - 7

The right qDevice is also subscribed to Enactment Group 50001. It will refer back to the unicast address of the sending qDevice straight away and then process the received enactment. This allows the initiating qDevice to manage repeat messages as necessary. The right qDevice Button 1 and Channel 1 are both subscribed to enactment 50001:10 and will thereby both process the received enactment message independently and transition to "on".

QDNE IN DETAIL

ENACTMENT FORMAT

At a conceptual level, an enactment is made up of two numbers: an "Enactment Group" and a "Selector".

The notation for an enactment is as follows:

$$\text{ENACTMENT} = \text{ENACTMENT GROUP} : \text{SELECTOR}$$

The **Enactment Group** is what connects all qDevices that share at least one common enactment. It is a number between 50000 and 65535. The Enactment Group also defines the UDP port the qDevice listens on.

The **Selector** identifies unique enactments within a given Enactment Group. It is a number between 1 and 255.

Enactment Group vs Selector

To understand the concept of an enactment being made up of both Enactment Groups and Selectors, please consider the following example:

- 3 qDevices – A, B and C
- A, B and C all share one common enactment (50003:9)
- A and B share two common enactments (50001:1 and 50001:2)

To achieve the required experiences two enactment groups are required:

QDEVICES	ENACTMENT GROUP	INDIVIDUAL ENACTMENTS
A, B, C	50003	50003:9 <ul style="list-style-type: none">• qDevice A – Button 3 / Channel 3• qDevice B – Button 3 / Channel 3• qDevice C – Button 1 / Channel 1
A, B	50001	50001:1 <ul style="list-style-type: none">• qDevice A – Button 1 / Channel 1• qDevice B – Button 1 / Channel 1 50001:2 <ul style="list-style-type: none">• qDevice A – Button 2 / Channel 2• qDevice B – Button 2 / Channel 2



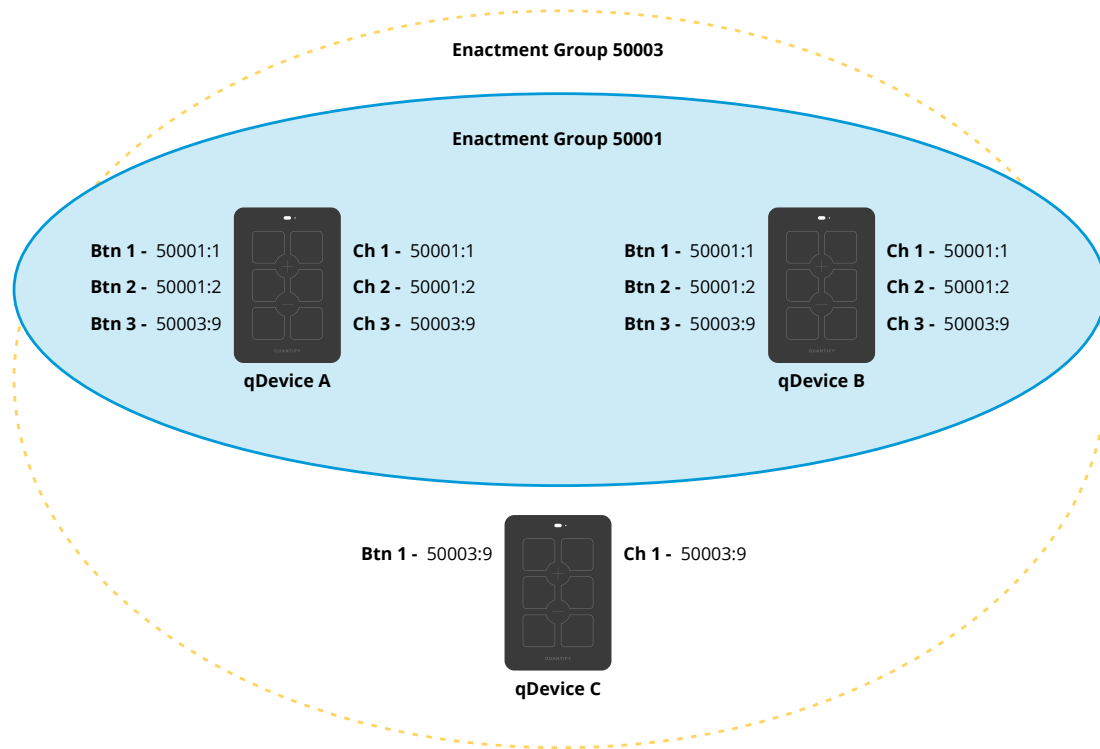


Figure 2

INFO: The Enactment Group and Selector scheme ensures that enactments are only processed by qDevices that have subscribed to that group, while allowing for multiple discrete experiences to co-exist within a given group of qDevices.

INFO: The Quantify Cloud ensures optimal Enactment Group allocation is generated for a given set of configured experiences.

ENACTMENT VALUE

Every enactment sent out also contains a value field. The value specifies the desired behaviour for receiving qDevices (on / off / dim etc).

VALUE	MEANING
0	Off
-3	Off Immediately (ignore any Run-On Timers if present)
1-99	Dim % or On (If Non-Dimmable)
100	On
103	Resume last Dim %

INFO: In the case of Scene enactments, the local qDevice uses its local configuration to overwrite the incoming value. This allows for a Scene enactment to trigger outputs to different states.



ACKNOWLEDGEMENTS

Acknowledgements are a fundamental part of the QDNE protocol. They improve reliability and ensure that all messages are received by relevant qDevices.

Enactments are acknowledged at an Enactment Group level. All devices subscribed to a given Enactment Group will send a unicast acknowledgement to the originating qDevice.

The example below demonstrates how devices C and D will generate acknowledgements for enactments in Enactment Group 50002, while devices B and E will drop them at the transport layer, as their UDP ports for that enactment aren't open.

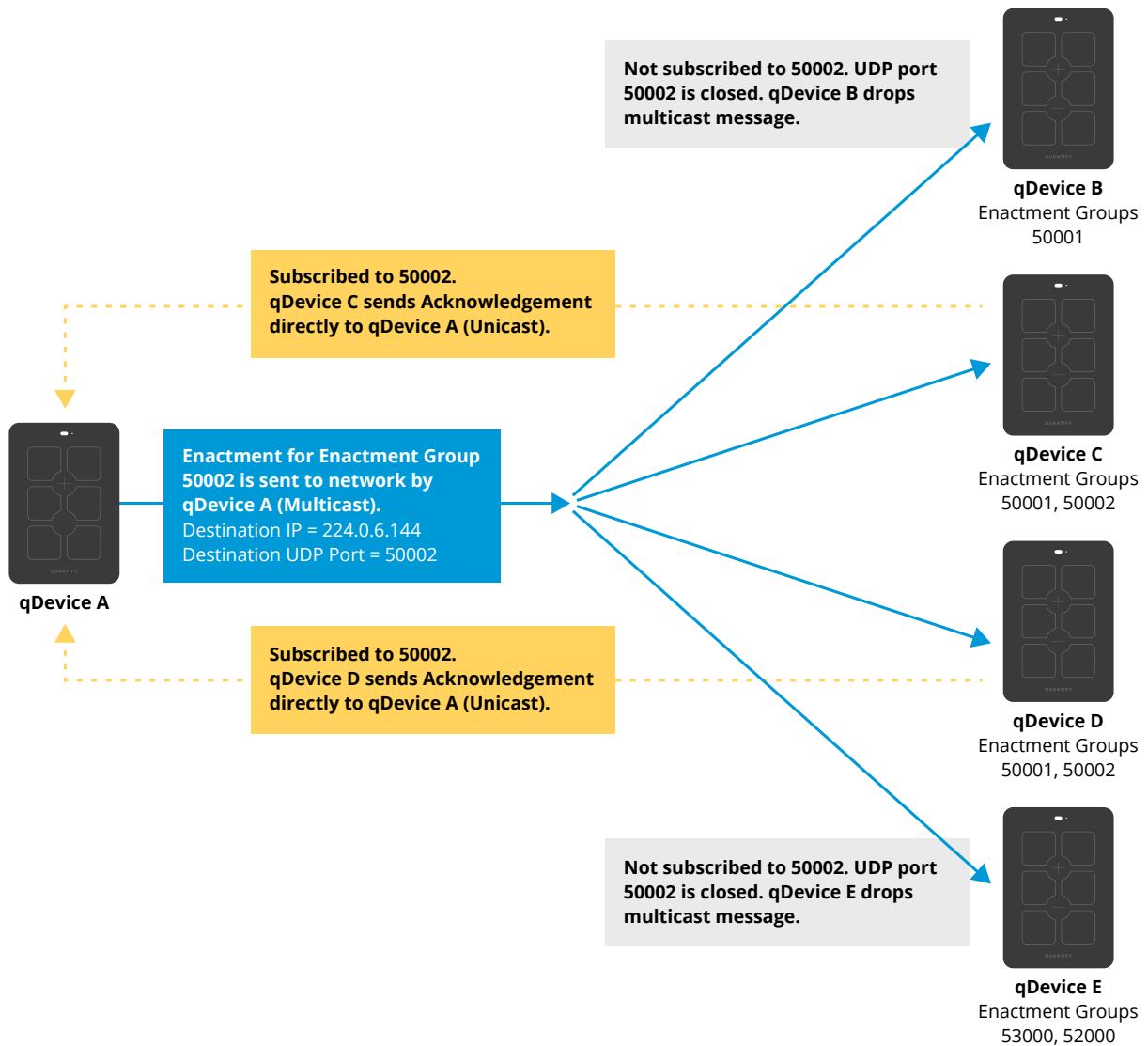


Figure 3

Each qDevice maintains a local partner count for the number of other qDevices it expects an acknowledgement from for any outgoing enactments. If the required number of unique acknowledgements is not reached within the timeout period, the outgoing enactment is resent. By design, enactments need to be idempotent.

INFO: A qDevice will retry an enactment a maximum of three times.



RETRY TIMERS

For version 3 firmware, the following retry timers are used:

Retry 1	1000 ms after initial enactment
Retry 2	2000 ms after Retry 1
Retry 3	1000 ms after Retry 2

For version 4 firmware, the following retry timers are used:

Retry 1	500 ms after initial enactment
Retry 2	500 ms after Retry 1
Retry 3	500 ms after Retry 2

MULTICAST HEARTBEAT

Because some wireless vendors do not implement multicast in their products in a robust manner, qDevices will sometimes be removed from the 224.0.6.144 multicast address without being notified by the network that this has occurred.

If qDevices haven't received a multicast message from the network for a randomised timeout between 6-10 seconds, it will send a heartbeat message out.

If a qDevice does not see a heartbeat message for 20 seconds, it will trigger a multicast message to bring the qDevice back into operation on the multicast address correctly.

OTHER QDNE MESSAGES

QDEVICE DISCOVERY

An application on the local network can discover online qDevices via a QDNE status message to the multicast address 224.0.6.144 on control port 49153.

All qDevices will respond with a unicast acknowledgement.

ENACTMENT DISCOVERY

An application can discover all available enactments on online qDevices through QDNE enactment discovery messages. All online qDevices will respond with the following information:

- Enactment Name
- Enactment - Enactment Group:Selector
- Type: - On / Off
 - Dimmable
 - Scene
 - Blind
 - Variable Blind

This allows third-party applications to quickly discover available services and interact with the Quantify ecosystem via QDNE.



FIRMWARE UPDATES

When used with the Quantify Cloud, firmware updates are directly performed via HTTPS and MQTT with each qDevice. QDNE supports the ability for one qDevice to share firmware updates with other qDevices, to reduce upstream bandwidth usage or comply with enterprise security policy.

MESSAGE SIGNING

QDNE messages support the cryptographic signing of messages to mitigate against replay attacks or undesirable enactment discovery. This is covered in more detail in the Security White Paper.





+61 (0)8 6254 0200

sales@quantifytechnology.com

Suite 2, 6 Brodie-Hall Drive, Bentley, Western Australia 6102

www.quantifytechnology.com