



CCNx 1.0 CCNmq Message Queue API Architectural Overview

Computer Science Laboratory
Networking & Distributed Systems

March 2014

CCNmq Message Queuing

Store and Forward Messaging for Applications

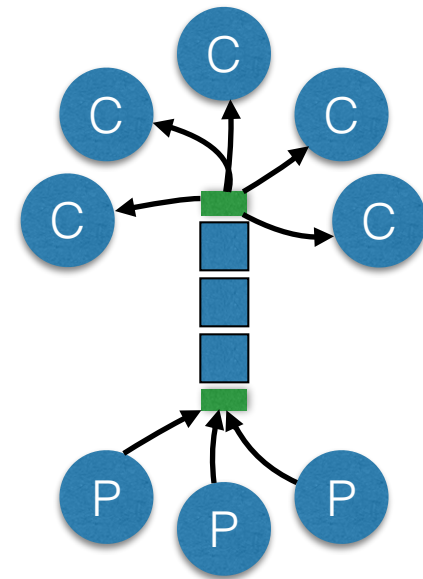
Producers enqueue Messages

Consumers dequeue Messages

Producers and Consumers are synchronous or asynchronous

Queues implement their respective semantics.

Not all queues must have the same semantics.



CCNmq Broker

Implementation Semantics

Enqueuing

Conditions that determine when a Message is enqueued. Confirmations, de-duplication, etc.

Dequeuing

Conditions that determine when a message is dequeued: In-order, at-least-one, at-most-one, exactly one, confirmations.

Security Policies

Authentication and authorization of publishers and consumers, privacy, etc.

Persistence

Mechanism used to implement the storage and lifetime of a Message

Availability

Fault-tolerance and recovery

Distribution

Coordination between multiple CcNmQ Brokers

CCNmq Producers

Transmit Messages to a named CCN message queue.

Participates with the message queue to enforce the enqueue policy.

In-order, at-least-one, at-most-one, exactly one, etc.

Duplicate Messages, Batching and coalescing.

CCNmq Consumers

Get Messages from a named CCN message queue.

Participates with the message queue to enforce the dequeue policy.

In-order, at-least-one, at-most-one, exactly one, etc.

Authentication and Authorization

Consumers may

CCNmq API

(draft)

Broker API

`ccnmq_Subscribe` Subscribe to future events.

`ccnmq_Unsubscribe` Stop a previous subscription.

`ccnmq_Publish` Publish an event.

`ccnmp_Identity` Represent the identity of a broker.

Queue API

`ccnmp_Enqueue`

`ccnmp_Dequeue`

CCNmq

Current Research

- Augment existing MQ protocols with the CCN security model.
- Abstract multiple MQ protocols enabling interoperability.
 - Names and naming, semantics, etc.
- Improve distribution performance via CCN caching.
- CCN primitives for fault tolerant services.
- CCMmq Queue discovery protocols.