

**Enterprise Computing
Einführung in das Betriebssystem z/OS**

**Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth**

WS2012/2013

WebSphere MQ Teil 2

Queues und Channels

Messages



Abb. 2.1

Eine Nachricht besteht aus zwei Teilen:

1. Message Header (Nachrichtenkopf),
2. Daten, die von einem Programm zu einem anderen geschickt werden.

Der Message Header beinhaltet in jedem Fall einen „Message Descriptor“ (MQMD), kann aber weitere Information beinhalten. Der Message Descriptor identifiziert die Nachricht (Message-ID) und enthält Steuerinformationen (auch als Attribute bezeichnet) wie Nachrichtentyp, Name der Target Queue, Expiration Time, Korrelations-ID, die Priorität und den Namen der lokalen Queue für eine mögliche Antwort.

Eine Nachricht kann bis zu 100 MByte lang sein.

Was ist eine Queue (Message Queue)

Die Begriffe "Queue" und "Message Queue" sind austauschbar.

Eine Queue ist eine Datenstruktur die verwendet wird, um Nachrichten zu speichern, bis sie von einer Anwendung abgefragt werden. Es ist einfach ein Speicherplatz der Nachrichten enthält. Die Nachrichten werden entweder von einem Anwendungsprogramm in die Queue gesetzt, oder durch einen Queue Manager als Teil seiner Operation.

Der Queue-Manager ist für die Queues verantwortlich, die er besitzt, für die Speicherung aller Nachrichten, die er empfängt, und das Auslesen von Nachrichten als Reaktion auf eine Anforderung eines Anwendungsprogramms.

Es gibt lokale Queues, die im Besitz des lokalen Queue-Managers sind, und entfernte (remote) Queues (z.B. eine Target Queue), die einem anderen Queue-Manager gehören.

Eine Queue ist ein benanntes Objekt (bis zu 48 Zeichen lang, so lang wie ein z/OS-Data-Set-Name), der von einem Queue Typ definiert wird. Der Queue Typ kann eine lokale Queue spezifizieren. Alternativ kann es eine lokale Definition einer Target-Queue sein, die sich auf einem entfernten Rechner mit einem entfernten Queue-Manager befindet.

Lokale Queues sind physische Queues; der Queue-Manager kann eine Nachricht in ihnen speichern. Ein Queue Manager behandelt nicht lokale Queues als Beschreibungen von Queues, die Definitionen enthalten, die durch einen entfernten lokalen Queue-Manager verwendet werden können.

Der MQPUT Befehl speichert eine Nachricht in der Transmission Queue. Es definiert auch in seinem Message Descriptor (MQMD) eine entfernte Target-Queue (die MQI Syntax nennt dies ein "Remote-Queue-Objekt"). Diese Informationen werden von dem lokalen Queue-Manager verwendet, um die Nachricht an die entsprechende Target Queue eines anderen entfernten Queue-Managers zu übertragen.

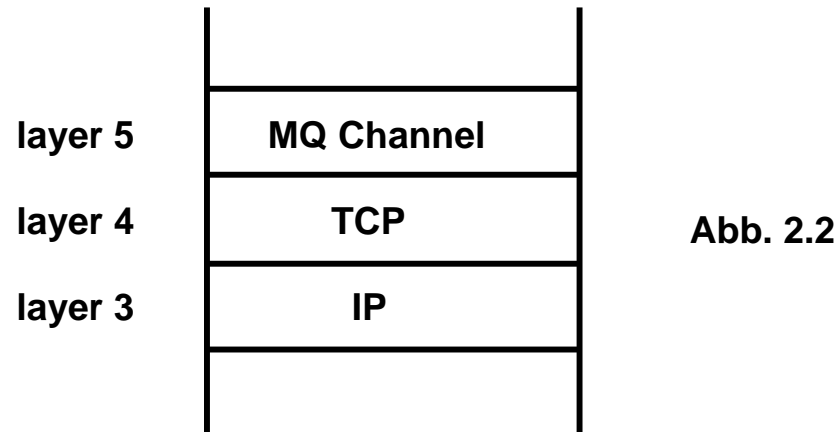


Abb. 2.2

Was ist ein Channel?

Alle Netzwerk-Kommunikation in WebSphere MQ wird über einen "**Channel**" (auch als "Message Channel" oder "MQ Channel" bezeichnet) durchgeführt. Insbesondere die Übertragung von Nachrichten von einer Transmission Queue an eine Target Queue eines entfernten Queue Managers erfolgt über einen Channel..

Ein Channel ist eine Punkt-zu-Punkt-Verbindung, die zwei verschiedene Queue-Manager miteinander verknüpft. Es ist eine logische Verbindung in der Schicht 5 des OSI-Modells. Telnet, 3270, SMTP, http und ftp sind weitere Beispiele für Schicht 5 Protokolle. Ein Channel wird durch ein sendendes und ein empfangendes- Programm realisiert, die über ein Netzwerk miteinander verbunden sind.

Channels verbergen die zugrundeliegenden Schicht 3 und Schicht 4 Kommunikationsprotokolle (zum Beispiel TCP und IP) vor den Anwendungen. Anwendungsprogrammierer brauchen nicht die physische Adresse des Programms zu kennen, dem sie eine Nachricht senden. Sie stellen ihre Nachrichten in einer Queue, und überlassen es dem Queue-Manager, sich um die Adresse des Ziel-Rechners zu kümmern, und darum, wie die Nachricht dorthin kommt. WebSphere MQ weiß, was zu tun ist, wenn das entfernte System nicht verfügbar ist oder das Zielprogramm nicht läuft oder beschäftigt ist.

Der Begriff "**Channel**" kann verschiedene Dinge in verschiedenen Umgebungen bedeuten. Eine MQ-Channel ist etwas ganz anderes als z.B. ein FICON-Channel.

MQ Message Channels

Wie funktioniert die Nachrichtenübertragung?

Jeder MQ Message Channel besteht aus zwei „Message Channel Agents“ (MCA), je einem auf dem sendenden und dem empfangenden Rechner. Jeder MCA übernimmt eine der folgenden Rollen:

- Der sendende MCA öffnet eine bestimmte Transmission Queue für exklusiven Input. Dies bedeutet, keine zwei Message Channels können konfiguriert werden, um Nachrichten von der gleichen Transmission Queue zu übernehmen. Der MCA erhält Nachrichten von einer bestimmten Transmission Queue und sendet sie an den Partner MCA.
- Der empfangende MCA empfängt Nachrichten von dem sendenden MCA. Für jede Nachricht, entfernt er den Transmission Queue Message Deskriptor aus der Nachricht und liest deren Inhalt. Er öffnet die Target Queue, die in dem Message Deskriptor für diese Nachricht angegeben ist und schreibt dann die Nachricht in die Target Queue.

Bitte beachten: Ein Channel ist mit einer einzigen Transmission Queue verbunden, kann aber mit mehreren Target Queues verbunden sein.

MQSeries benutzt den Begriff „Objekt“, um damit Queues und Channels zu bezeichnen.

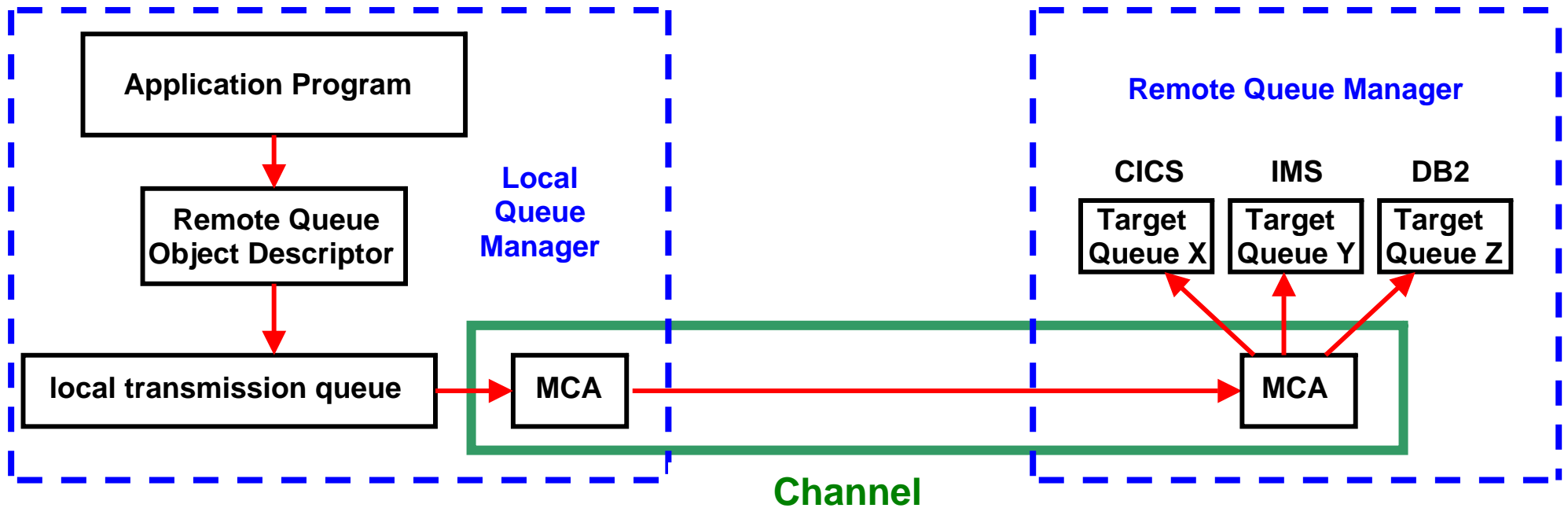


Abb. 2.3

Message Channel Agents

Ein **Message Channel Agent** (MCA) ist ein Programm, welches das Senden und Empfangen von Nachrichten steuert. Es gibt einen MCA an jedem Ende eines Channels. Ein MCA nimmt Nachrichten aus der Transmission Queue und legt sie auf die Kommunikationsverbindung. Der andere MCA empfängt Nachrichten und liefert sie an eine Target Queue des entfernten Queue Managers. Einem MQ-Channel wird durch ein Paar von MCAs und das Kommunikationsnetz dazwischen implementiert.

Um Nachrichten von einem lokalen Queue-Manager an einen entfernten Queue-Manager, zu schicken, muss der lokale Queue Manager eine Kommunikationsverbindung zum entfernten Queue Manager aufsetzen (zum Beispiel mit dem MQCONNECT Befehl). Der empfangende MCA muss auf dem entfernten Queue-Manager gestartet sein, um Nachrichten über die Kommunikationsverbindung zu empfangen. Diese Einweg-Verbindung, bestehend aus dem sendenden MCA, der Kommunikationsverbindung, und dem empfangenden MCA, wird als Channel bezeichnet. Der sendende MCA übernimmt Nachrichten von einer Transmission Queue und sendet sie über einen Channel an den empfangenden MCA. Dieser empfängt die Nachrichten und legt sie auf eine Target Queue.

Das Remote Queue Object Descriptor (MQOD) enthält die Message Descriptor Information, die für den Transmission Queue Header benötigt wird.

MQ-Kanäle sind unidirektional. Für die bidirektionale Kommunikation müssen zwei Channels (ein Channel Paar) bestehend aus einem Sender Channel und einem Empfänger Channel definiert werden.

Target Queue Definition

Wenn Sie ein WebSphere MQ-Queue öffnen, führt der MQOPEN Befehl eine Namensauflösung durch

Die MQPUT Kommando eines Anwendungsprogramme schreibt eine Nachricht in einer virtuellen Remote-Queue, die nur eine Definition einer Target Queue auf einem entfernten Rechner darstellt (remote Queue-Definition). Es ist die Aufgabe des lokalen Queue-Managers, die Nachricht in eine lokale Transmission Queue zu laden, und an die richtige Target Queue weiterzuleiten. Dieses Target Queue ist lokal für einen einem entfernten Queue-Manager auf einem Remote System.

Auf dem entfernten System (und seine Queue-Manager) können mehrere Target Queues existieren, die mit unterschiedlichen Anwendungsprogrammen assoziiert sind, zum Beispiel einer CICS-Transaktion, einer IMS-Transaktion oder einer DB2 Stored Procedure.

Eine einzelne Transmission Queue und ein einzelner MQ Message Channel kann mehrere Target Queues auf einem bestimmten Remote-System bedienen. Der Message Deskriptor in der Nachricht, die an das entfernte System gesendet wird, definiert die Target Queue, welche die Nachricht empfangen soll.

Wie in der folgenden Abbildung dargestellt, kann ein lokales System gleichzeitige mehrere WebSphere MQ Verbindungen zu mehreren Remote-Systemen aufrechterhalten. Jede Verbindung benötigt eine separate Transmission Queue und den damit verbundenen MQ Message Channel. Die MQPUT Kommando eines Anwendungsprogramme schreibt eine Nachricht an eine von mehreren entfernten Target Queues. Der lokale Queue-Manager stellt die Nachricht in die entsprechende Transmission Queue. Damit wird die Nachricht automatisch an den richtigen Remote Queue Manager ausgeliefert.

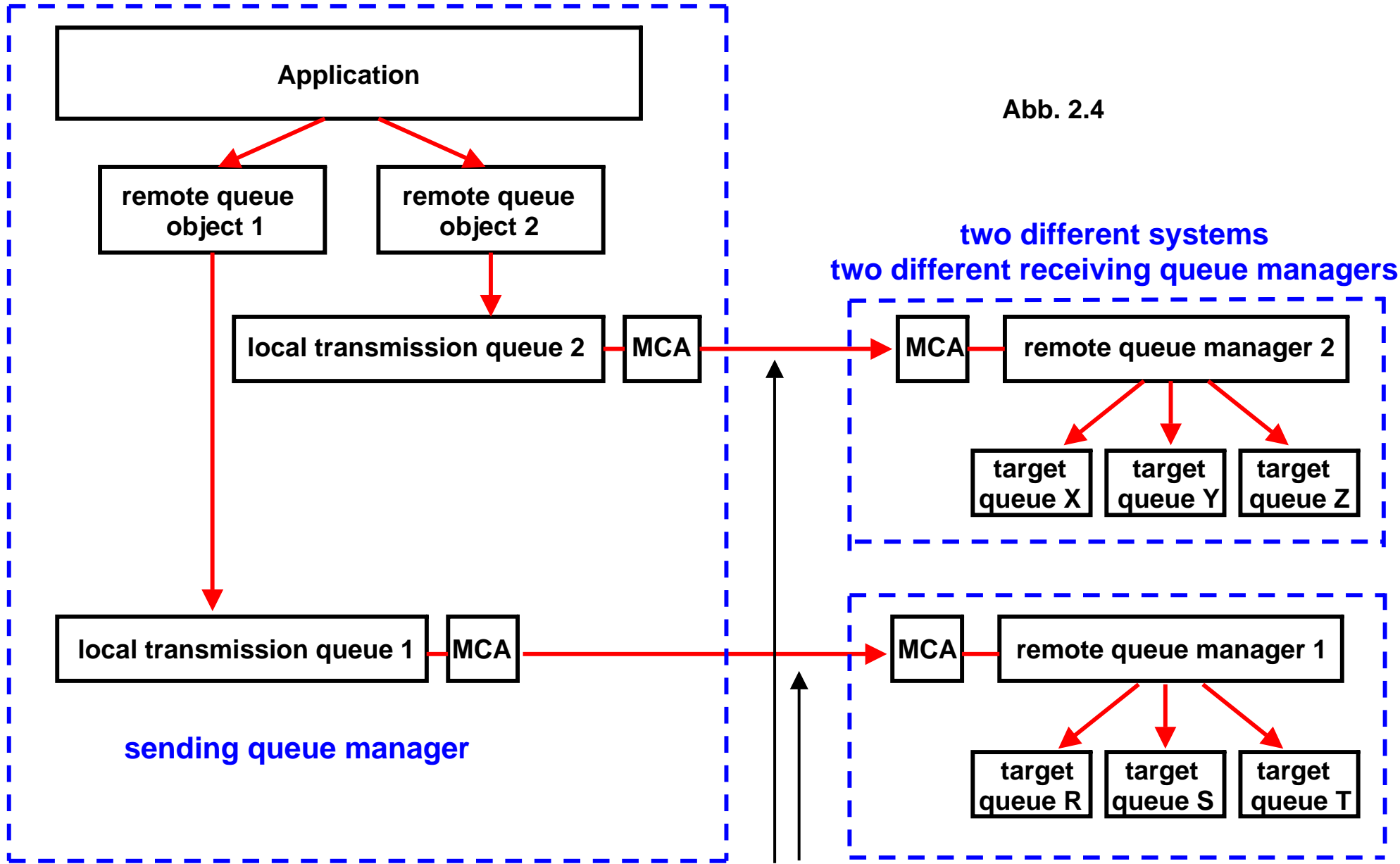


Abb. 2.4

Two different message channels

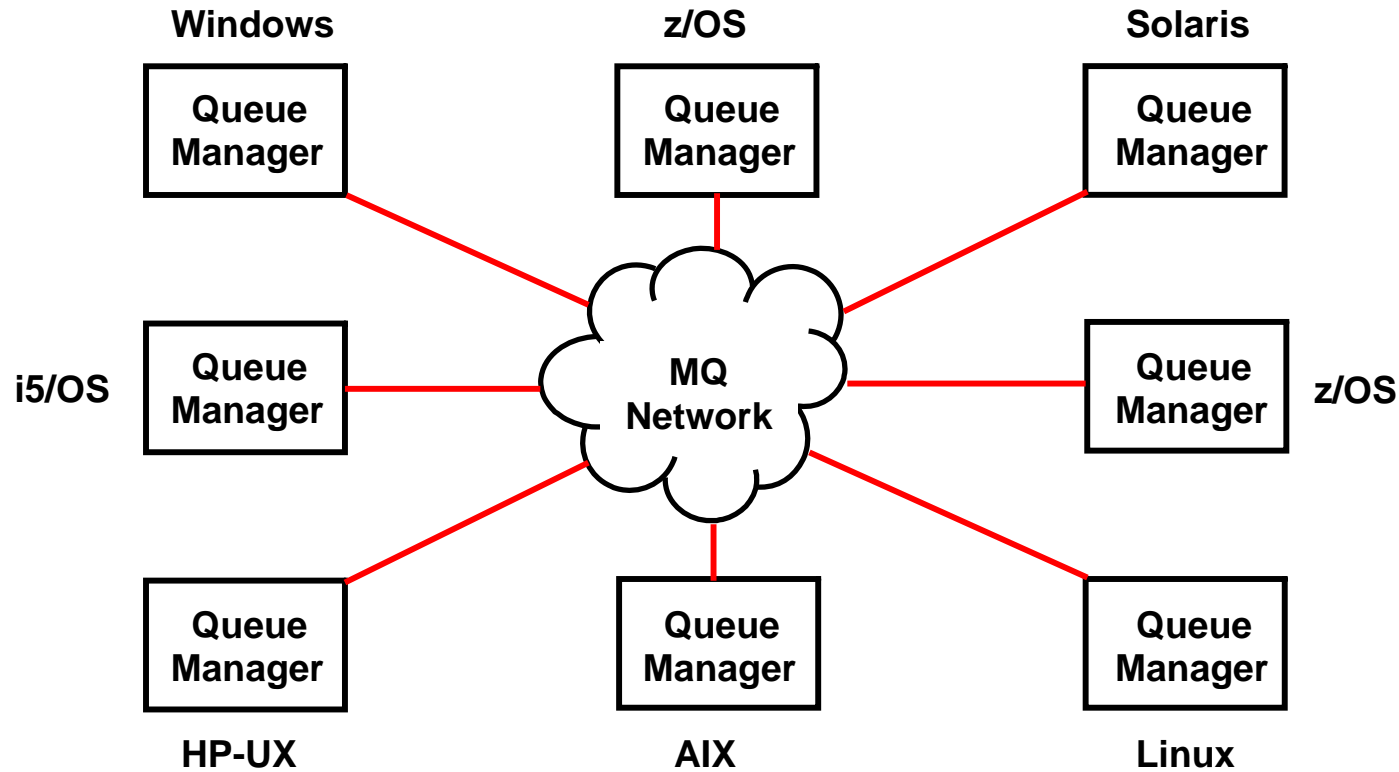


Abb. 2.5

WebSphere MQ Netzwerk

Alle Queue Manager in einem Netzwerk von WebSphere MQ Systeme müssen eindeutige Namen haben. Sie ordnen die Namen den Queue Managern zu, wenn Sie das WebSphere MQ-Netz definieren und installiere, genau so, wie Sie IP-Adressen definieren, wenn Sie eine IP-Netzwerk aufzubauen.

Denken Sie daran, genauso wie bei CICS oder Apache, WebSphere MQ ist es egal, auf welcher Betriebssystem-Plattform es läuft.

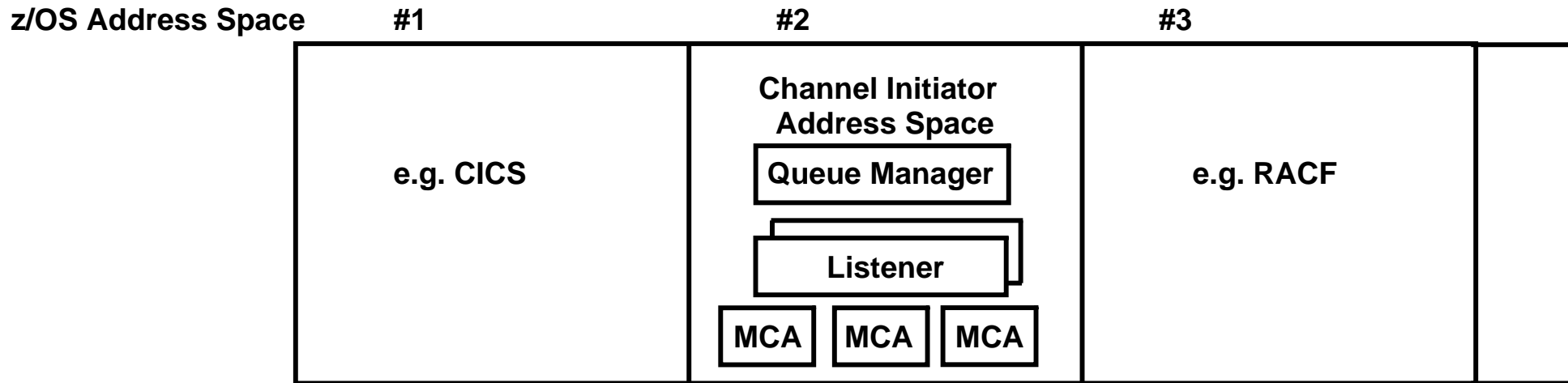


Abb. 2.6

z/OS Channel Initiator

Unter z/OS läuft WebSphere MQ als z/OS-Subsystem in seinen eigenen Adressraum (Adressraum # 2 in dem obigen Beispiel). Innerhalb des Subsystems wird der Queue-Manager durch das Ausführen eines JCL-Prozedur gestartet. Diese spezifiziert die z/OS-Datasets, die Informationen über die Log Files, die Object Definitionen und Message Data (page sets) enthalten. Alle Queue-Manager in Ihrem Netzwerk müssen eindeutige Namen haben, auch wenn sie auf verschiedenen Systemen und Plattformen laufen.

Ein Channel Initiator ist ein Bestandteil eines Queue-Managers. In z/OS existiert ein Channel Initiator für jeden Queue-Manager (es können mehrere Queue-Manager in getrennten Regionen existieren). Der Channel Initiator läuft zusammen mit seinen Queue-Manager innerhalb desselben z/OS Adressenraums. Er beherbergt alle Message Channel Agents (MCAs) für einen Queue-Manager. Tausende von MCA Prozessen können innerhalb des Channels Initiators gleichzeitig laufen. Der Channel Initiator überwacht die vom System definierte Queue SYSTEM.CHANNEL.INITQ, die Initiation Queue für eine Transmission Queue (siehe hierzu den folgenden Teil 3 dieses Kapitels).

Sie können den Channel Initiator benutzen, um Channels zu starten.

Listener

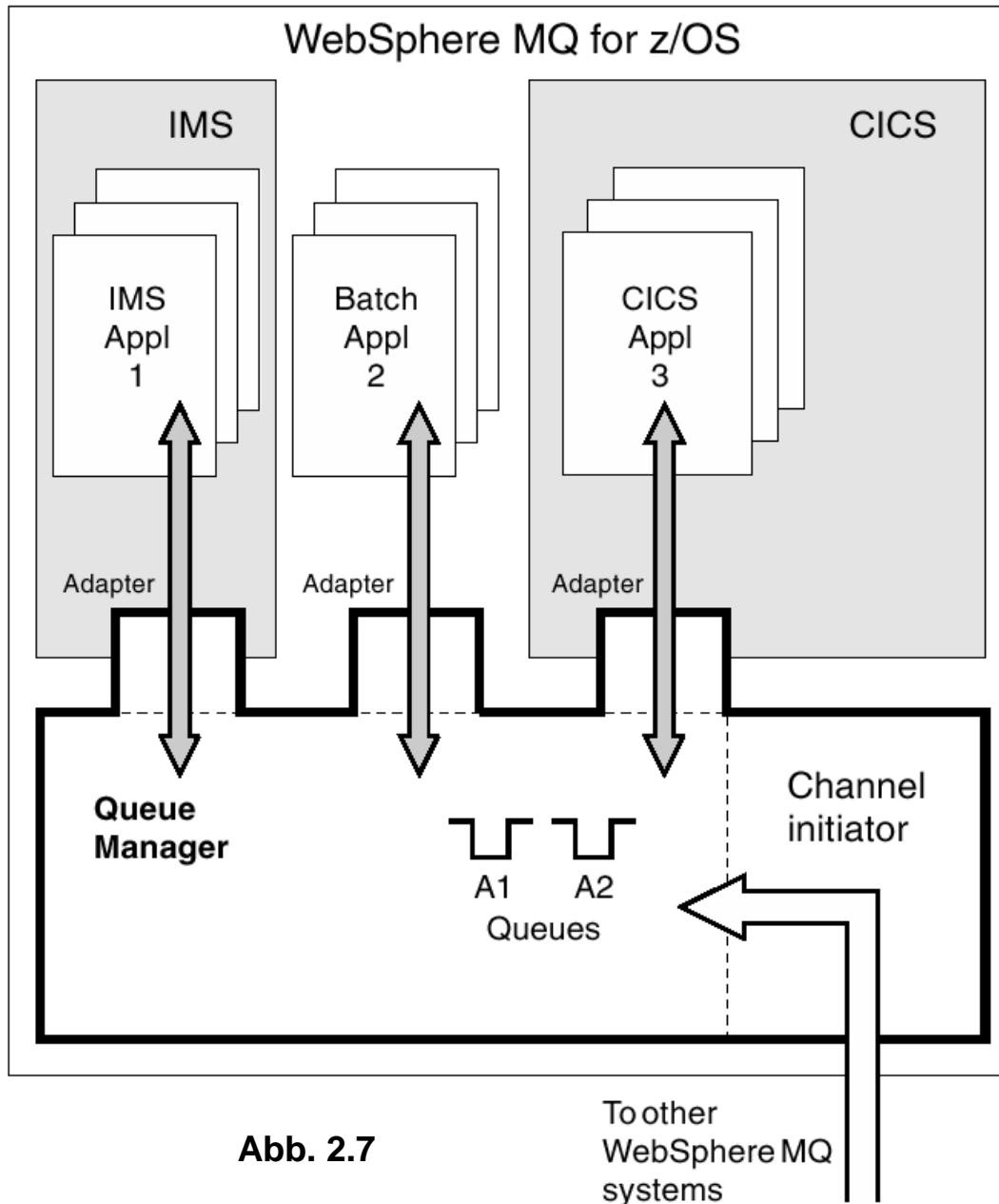
Um Nachrichten zu empfangen, muss ein Listener-Programm auf der Empfängerseite gestartet worden sein.

Das Abhören (monitor) von TCP/IP-Ports geschieht mit Hilfe des WebSphere MQ für z/OS Channel-Initiators. Der Listener, ein WebSphereMQ Komponente, und ein Teil des empfangenden Queuemangers, überwacht Nachrichten, die auf einem TCP/IP-Port ankommen. Er erstellt einen Message Channel Agent (MCA), der diese Verbindung verarbeitet. Wenn eine Nachricht eintrifft, startet er den Message Channel Agent. Die MCA speichert die Nachricht in die Target-Queue, die in dem Message Descriptor angegeben ist.

Port 1414 ist der Standard TCP/IP Port für WebSphere MQ. Mehrere TCP/IP Listener mit verschiedenen Ports können innerhalb des Channels Initiators gestartet werden. Jeder Listener hört einem bestimmten TCP/IP-Port ab (monitors the Port). Ein Listener wird mit der START LISTENER Befehl innerhalb des Queue-Manager Subsystems gestartet.

Zusätzlich zu den üblichen TCP/IP Listeners unterstützt WebSphere MQ für z/OS auch SNA LU 6.2 Listener.

Das Anwendungsprogramm, das die eingehende Nachricht verarbeiten soll, kann manuell oder automatisch gestartet werden. Um das Programm automatisch zu starten, muss eine Initiation Queue und ein Anwendungsprogramm mit der lokalen Queue assoziiert werden, und der Trigger-Monitor muss laufen. Dies wird in Teil 3 dieses Moduls diskutiert.



WebSphere MQ auf dem gleichen z/OS System

MQ Kommunikation wird in der Regel zwischen entfernten Systemen durchgeführt. Es kann aber auch für eine Kommunikation zwischen zwei Regions auf dem gleichen z/OS System eingesetzt werden. Die Abbildung auf der linken Seite ist ein Beispiel für eine Programm-zu-Programm-Kommunikation in einem einzigen z/OS-System, zum Beispiel zwischen einer CICS Region und einer IMS-Region.

Eine Übungsaufgabe (Tutorial) für dieses Szenario läuft auf unserem Rechner. Es ist zu finden unter

<http://www.informatik.uni-leipzig.de/cs/Literature/VorlesRef/TutorialMQ1.pdf>

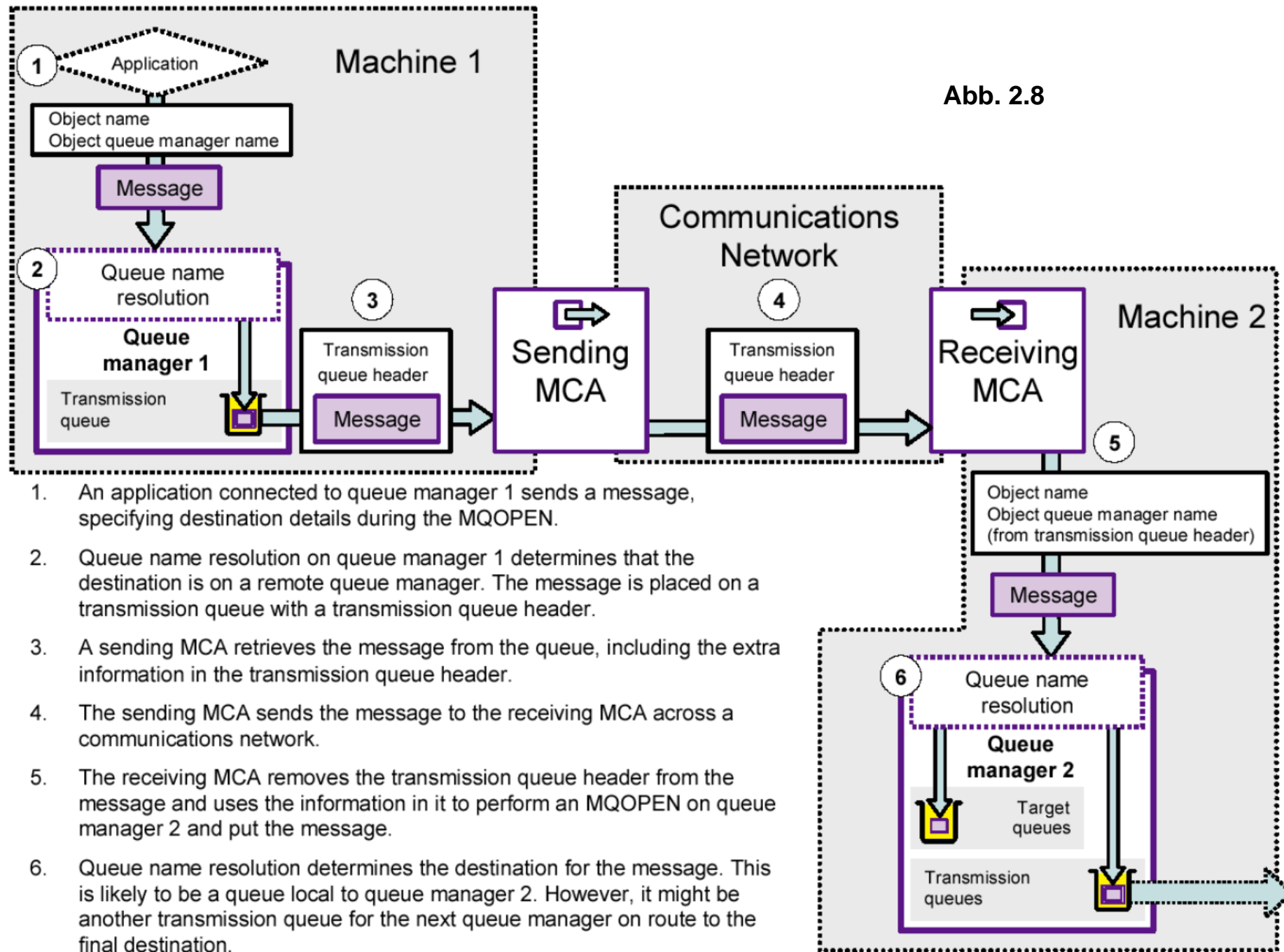
MQPUT

Beim Ausführen eines MQPUT Calls übergibt das Anwendungsprogramm neben der Nachricht einen Remote Queue Object Descriptor (MQOD) an den lokalen Queue Manager. Dieser enthält spezifisch den Namen des Entfernten Queue Managers und den Namen der entfernten Target Queue.

Während der Queue Namensauflösung durch den sendenden Queue-Manager, (wo der sendende MCA läuft), generiert der lokale Queue Manager hieraus den Message Descriptor, der in den Message Header (Transmission Queue Header) eingebaut wird. Er wird verwendet, um die Nachricht an den Queue-Manager zu übertragen, auf dem der empfangende MCA läuft. Dieser schreibt die Nachricht auf eine von mehreren Target Queues, die von dem empfangenden Queue-Manager gesteuert werden. Der Transmission Queue Header enthält die folgenden Informationen:

- **Remote Queue-Name:**
Der Name der Target Queue für die Nachricht, die von der Queue-Manager während der Queue Namensauflösung ermittelt wurde. Wenn der MCA im entfernten Queue-Manager die Nachricht in eine Queue zu schreiben versucht, entnimmt er diesen Namen der Queue dem MQ Object Descriptor (MQOD) beim Öffnen der Queue. MQOD ist eine Struktur (in C++) oder ein Copybook (in Cobol), die als Parameter in dem MQOPEN Anruf des sendenden Anwendungsprogramms benutzt wurde.
- **Remote Queue Manager Name:**
Dies ist der Name des Target-Queue-Managers, der die Target Queue hostet. Dies ist möglicherweise nicht der Name des Queue Managers, der die Nachricht empfängt. Dieser Fall kann auftreten, wenn dieser Queue Manager nicht die Final Destination für die Nachricht ist. Eine MQ Channel Verbindung kann über ein Netzwerk von Queue Managen zu dem empfangenden Queue Manager erstellt werden.

Abb. 2.8



1. An application connected to queue manager 1 sends a message, specifying destination details during the MQOPEN.
2. Queue name resolution on queue manager 1 determines that the destination is on a remote queue manager. The message is placed on a transmission queue with a transmission queue header.
3. A sending MCA retrieves the message from the queue, including the extra information in the transmission queue header.
4. The sending MCA sends the message to the receiving MCA across a communications network.
5. The receiving MCA removes the transmission queue header from the message and uses the information in it to perform an MQOPEN on queue manager 2 and put the message.
6. Queue name resolution determines the destination for the message. This is likely to be a queue local to queue manager 2. However, it might be another transmission queue for the next queue manager on route to the final destination.