Ultra Messaging (Version 6.13.1)

Dynamic Routing Guide

Contents

Chapter 1

Introduction

This document explains design concepts and product implementation for the Ultra Messaging *Dynamic Routing Option* (DRO).

For policies and procedures related to Ultra Messaging Technical Support, see UM Support.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided.
INFORMATICA LLC PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

The Ultra Messaging Dynamic Routing Option (DRO) consists of a daemon named "tnwgd" that bridges disjoint **Topic Resolution Domains** (TRDs) by effectively forwarding control and user traffic between them. Thus, the DRO facilitates WAN routing where multicast routing capability is absent, possibly due to technical obstacles or enterprise policies.

FYI: for historical reasons, the DRO has gone by several names:

- Gateway
- tnwg = "Twenty Nine West Gateway"
- UM Router
- Dynamic Router
- DRO = Dynamic Routing Option

In the UM documentation, the term "DRO" is generally used for brevity, but sometimes various abbreviations that include "tnwg" are used.

1.1 DRO Features

The DRO includes the following features:

- Full bidirectional forwarding
- Multi-hop forwarding
- Mesh, loop, or alternate path DRO configurations
- Automatic rerouting around faults
- Support for wildcard receivers
- Support of Request/Response messages
- Traffic filtering on multiple criteria
- DRO resilience
- UMP persistence support
- UM transport monitoring statistics
- Web Monitoring
- MIM and UIM forwarding

The following features are not fully supported in this release of the DRO:

- Queuing, both ULB and Brokered (including brokered JMS)
- Multitransport Threads (MTT)

If you desire any of these features or any configuration or topology not presented in this document, please contact Informatica Ultra Messaging Support for possible alternatives.

Note

The DRO is not directly supported on the OpenVMS platform. UM applications running on the OpenVMS platform, however, can use a DRO running on a different platform, such as Microsoft Windows or Linux.

Chapter 2

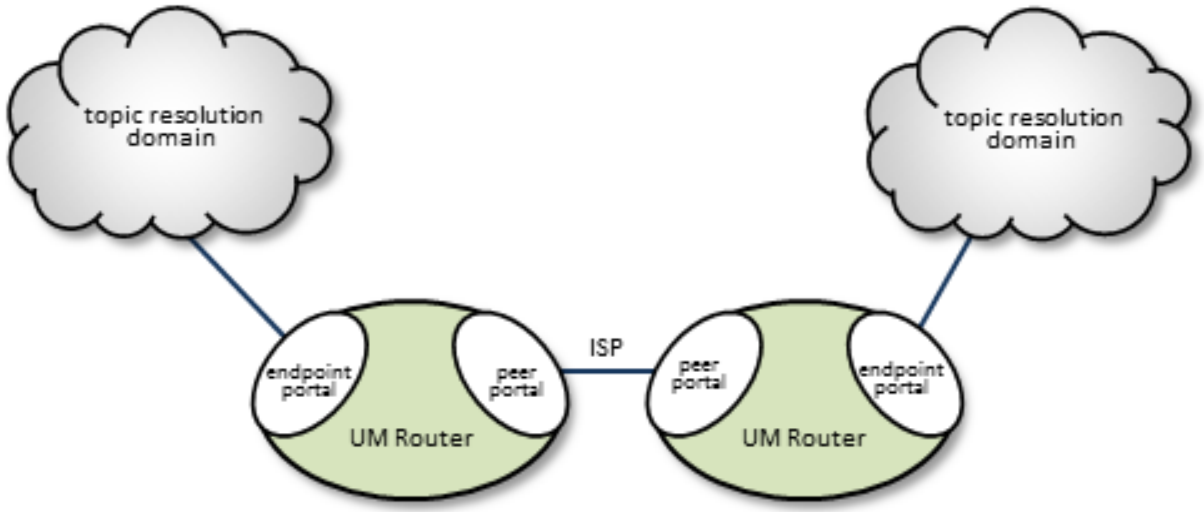
DRO Architecture

2.1 DRO Portals

The DRO uses interfaces, called portals, through which to pass data. A DRO consists of two or more bidirectional portals that may be one of two types:

- An endpoint portal, which communicates directly to a UM topic resolution domain (TRD; see Topic Resolution Domains).
- A peer portal, which communicates via TCP with another peer portal (of another DRO), allowing tunneling between DROs. Two peer portals connected to each other are referred to as companion peers, and utilize TCP connections for all data and control traffic (UDP is not supported for this). Compression and encryption can be applied to peer links.

The figure below shows a simple DRO use case, where two DROs bridge an ISP to connect two TRDs.



You configure portals in the DRO's XML configuration file, specifying the portal's name, cost, UM Configuration, Access Control Lists and other attributes. See XML Configuration Reference.

2.2 Topic Resolution Domains

Since topic resolution uses UDP, sources and receivers must have UDP connectivity to each other. When they do, we consider them to be in the same topic resolution domain (TRD). More specifically, UM contexts must satisfy the following two requirements to belong to the same topic resolution domain.

- The contexts must use the same topic resolution UM configuration (i.e., resolver_∗ options are the same).
- Contexts can communicate using the protocols required for both message transport and topic resolution traffic.

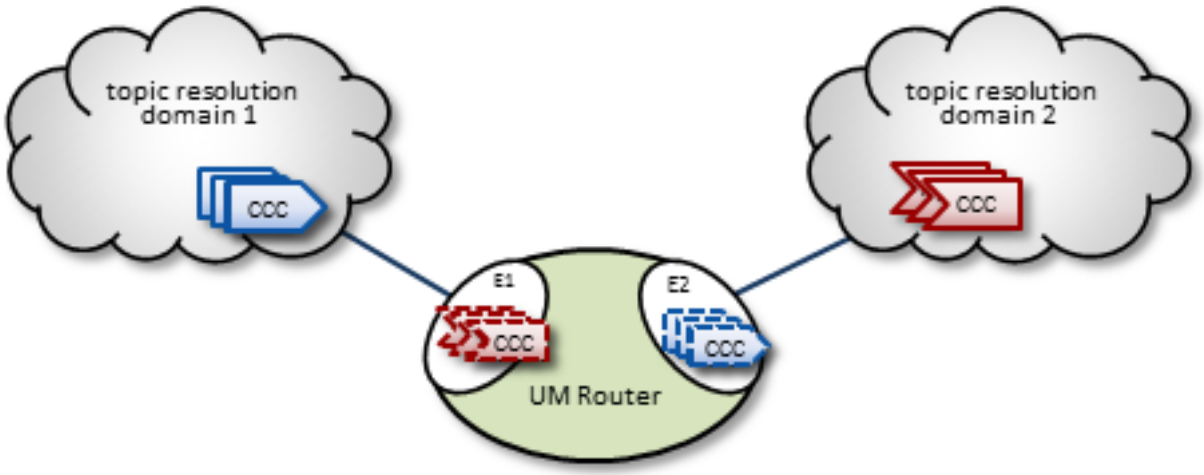
For example, two contexts on separate machines in the same LAN are not in the same topic resolution domain if they use different resolver addresses. See Multicast Resolver Network Options. A topic resolution domain can span a WAN if the UM contexts on each side of a firewall use the same UM configuration and the firewall allows UDP traffic (multicast or unicast) to pass.

Each endpoint portal must identify its associated topic resolution domain with a domain-id the DRO's XML configuration file, as in the example below. All portals in the same TRD must have the same domain-id, and different TRDs networked together via DROs must have domain-ids unique to each other.

```
<portals>
  <endpoint>
    <name>LAN100</name>
    <domain-id>100</domain-id>
    <lbm-config>lan100.cfg</lbm-config>
  </endpoint>
  <endpoint>
    <name>LAN200</name>
    <domain-id>200</domain-id>
    <lbm-config>lan200.cfg</lbm-config>
  </endpoint>
</portals>
```

2.3 Proxy Sources and Proxy Receivers

To resolve a topic across a DRO (described in Basic DRO Operation), the DRO creates, within portals, proxy sources and proxy receivers (shown in the figure below by their dashed lines). These proxies behave like their UM counterparts; they resolve topics on the TRDs like normal sources and receivers, and the DRO internally passes data from one portal to the other. However unlike regular sources, proxy sources do not have retransmission retention buffers normally used for Late Join or OTR.



Portals exist while the DRO is running, however, the DRO creates proxy sources and receivers during topic resolution and deletes them when the topic is retired.

Note

The proxy sources created by the DRO are unrelated to proxy sources created by the UMP persistent store.

2.3.1 DRO and Transport Sessions

When the DRO creates proxy receivers to get messages to forward, be aware that the *transport sessions* carrying those messages are not extended to the destination TRD. Instead, the proxy receiver simply takes the messages from the originating transport sessions and transfers them to the destination DRO's proxy sources. Those proxy sources create new transport sessions for those outgoing messages.

The proxy sources' outgoing transport sessions are unrelated to the originating sources' transport sessions. They can even use different transport types, performing a protocol conversion. In fact, a single transport session can contain multiple sources from different originating publishing applications for the same topic. Alternatively, multiple sources from the same originating publishing application which are mapped to the same originating transport session can be split into multiple transport sessions by the proxy sources in a remote TRD.

One consequence of the independence of incoming and outgoing transport sessions is that TCP flow control does not transit the DRO. A slow receiver in a remote TRD cannot "push back" on a fast source. In cases where a TCP transport session is slowed down due to one or more slow receivers, an intermediate DRO will eventually have to drop messages.

Warning

A single source's "source string" will be different in different TRDs connected by DROs. See **Source Strings in a Routed Network** for details.

2.4 Routing

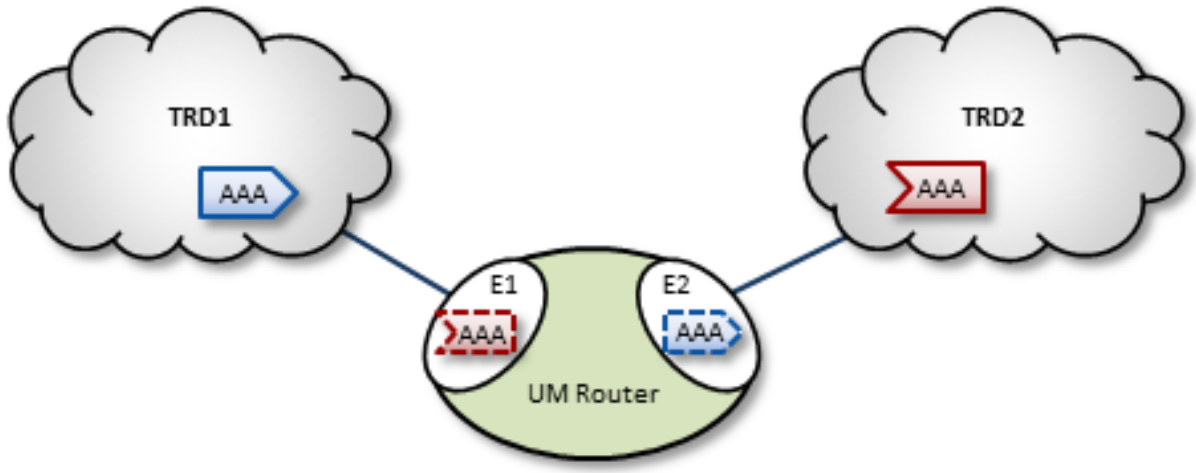
In multiple-DRO environments where more than one DRO can provide possible messaging pathways, the DROs are able to cooperatively determine and establish optimal routes. Also, the DRO network is able to detect link or other DRO outages and automatically reroute traffic as needed. See Routing Topologies for more information.

Chapter 3

DRO Concepts

3.1 Basic DRO Operation

The diagram below shows a DRO bridging topic resolution domains TRD1 and TRD2, for topic AAA, in a direct link configuration. Endpoint E1 contains a proxy receiver for topic AAA and endpoint E2 has a proxy source for topic AAA.



To establish topic resolution in an already-running DRO, the following sequence typically occurs in an example like the above figure.

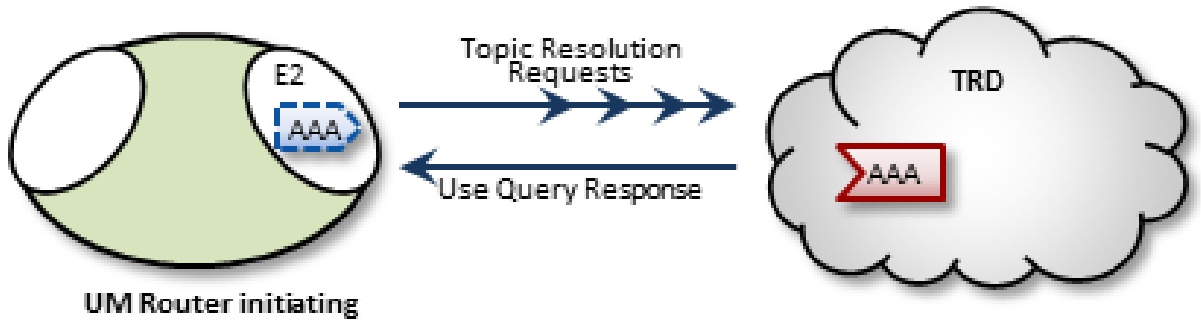
1. A receiver in TRD2 issues a TQR (Topic Query Record) for topic AAA.
2. Portal E2 receives the TQR and passes information about topic AAA to all other portals in the DRO. (In this case, E1 is the only other portal.)
3. E1 immediately responds with three actions: a) create a proxy receiver for topic AAA, b) the new proxy receiver sends a TQR for AAA into TRD1, and c) E1 issues a Topic Interest message into TRD1 for the benefit of any other DROs that may be connected to that domain.
4. A source for topic AAA in TRD1 sees the TQR and issues a TIR (Topic Information Record).

5. E2 creates proxy source AAA, which then issues a TIR to TRD2. The receiver in TRD2 joins the transport, thus completing topic resolution.

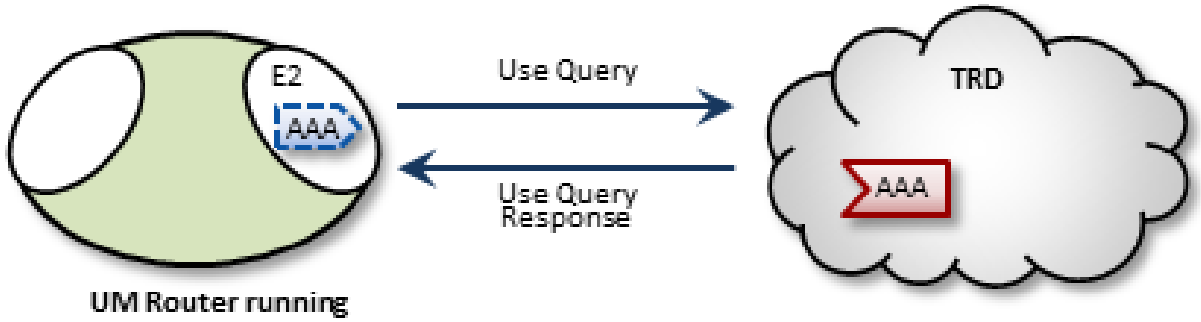
6. E1's AAA proxy receiver sees the TIR and requests that E2 (and any other interested portals in the DRO, if there were any) create a proxy source for AAA.

3.1.1 Interest and Use Queries

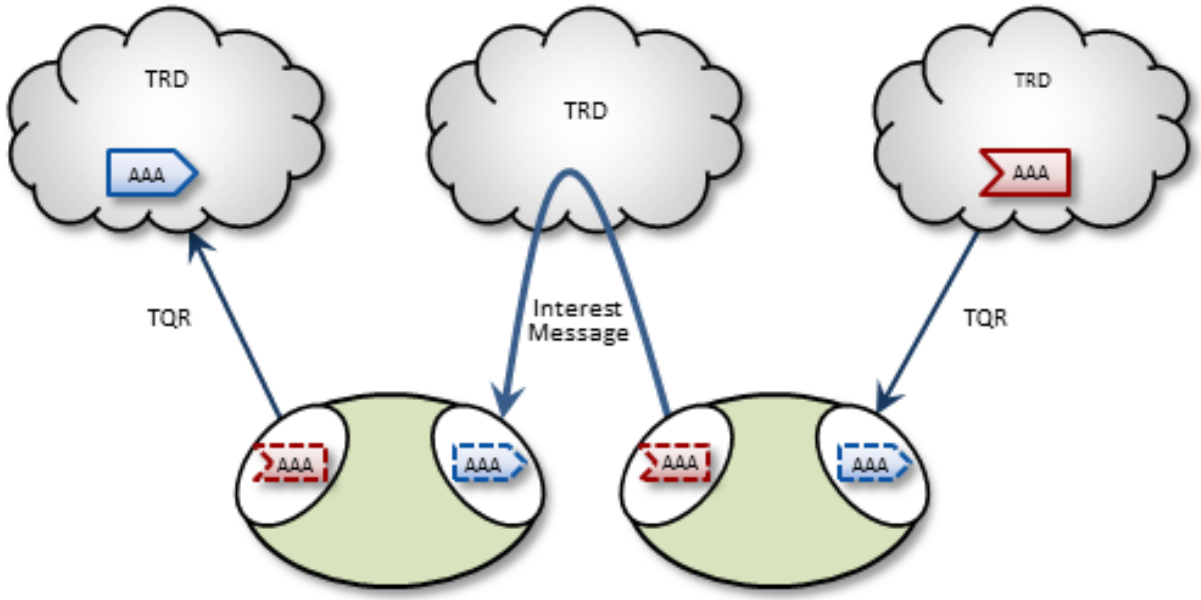
When a DRO starts, its endpoint portals issue a brief series of Topic Resolution Request messages to their respective topic resolution domains. This provokes quiescent receivers (and wildcard receivers) into sending Use Query Responses, indicating interest in various topics. Each portal then records this interest.



After a DRO has been running, endpoint portals issue periodic Topic Use Queries and Pattern Use Queries (collectively referred to as simply Use Queries). Use Query Responses from UM contexts confirm that the receivers for these topics indeed still exist, thus maintaining these topics on the interest list. Autonomous TQRs also refresh interest and have the effect of suppressing the generation of Use Queries.



In the case of multi-hop DRO configurations, DROs cannot detect interest for remote contexts via Use Queries or TQRs. They do this instead via Interest Messages. An endpoint portal generates periodic interest messages, which are picked up by adjacent DROs (i.e., the next hop over), at which time interest is refreshed.



You can adjust intervals, limits, and durations for these topic resolution and interest mechanisms via DRO configuration options (see XML Configuration Reference).

3.1.2 DRO Keepalive

To maintain a reliable connection, peer portals exchange DRO Keepalive signals. Keepalive intervals and connection timeouts are configurable on a per-portal basis. You can also set the DRO to send keepalives only when traffic is idle, which is the default condition. When both traffic and keepalives go silent at a portal ingress, the portal considers the connection lost and disconnects the TCP link. After the disconnect, the portal tries to reconnect. See <gateway-keepalive>.

3.1.3 Final Advertisements

DRO proxy sources on endpoint portals, when deleted, send out a series of final advertisements. A final advertisement tells any receivers, including proxy receivers on other DROs, that the particular source has gone away. This triggers EOS and clean-up activities on the receiver relative to that specific source, which causes the receiver to begin querying according to its topic resolution configuration for the sustaining phase of querying.

In short, final advertisements announce earlier detection of a source that has gone away, instead of transport timeout. This causes a faster transition to an alternative proxy source on a different DRO if there is a change in the routing path.

3.1.4 More About Proxy Sources and Receivers

The domain-id is used by Interest Messages and other internal and DRO-to-DRO traffic to ensure forwarding of all messages (payload and topic resolution) to the correct recipients. This also has the effect of not creating proxy sources/receivers where they are not needed. Thus, DROs create proxy sources and receivers based solely on receiver interest.

If more than one source sends on a given topic, the receiving portal's single proxy receiver for that topic receives all messages sent on that topic. The sending portal, however creates a proxy source for every source sending on the topic. The DRO maintains a table of proxy sources, each keyed by an Originating Transport ID (OTID), enabling the proxy receiver to forward each message to the correct proxy source. An OTID uniquely identifies a source's transport session, and is included in topic advertisements.

3.1.5 Protocol Conversion

When an application creates a source, it is configured to use one of the UM transport types. When a DRO is deployed, the proxy sources are also configured to use one of the UM transport types. Although users often use the same transport type for sources and proxy sources, this is not necessary. When different transport types are configured for source and proxy source, the DRO is performing a protocol conversion.

When this is done, it is very important to configure the transports to use the same maximum datagram size. If you don't, the DRO can drop messages which cannot be recovered through normal means. For example, a source in TRD1 can be configured for TCP, which has a default maximum datagram size of 65536. If a DRO's remote portal is configured to create LBT-RU proxy sources, that has a default maximum datagram size of 8192. If the source sends a user message of 10K, the TCP source will send it as a single fragment. The DRO will receive it and will attempt to forward it on an LBT-RU proxy source, but the 10K fragment is too large for LBT-RU's maximum datagram size, so the message will be dropped.

The solution is to override the default maximum datagram sizes to be the same. Informatica generally does not recommend configuring UDP-based transports for datagram sizes above 8K, so it is advisable to set the maximum datagram sizes of all transport types to 8192, like this:

```
context transport_tcp_datagram_max_size 8192
context transport_lbtrm_datagram_max_size 8192
context transport_lbtru_datagram_max_size 8192
context transport_lbtipc_datagram_max_size 8192
source transport_lbtsmx_datagram_max_size 8192
```

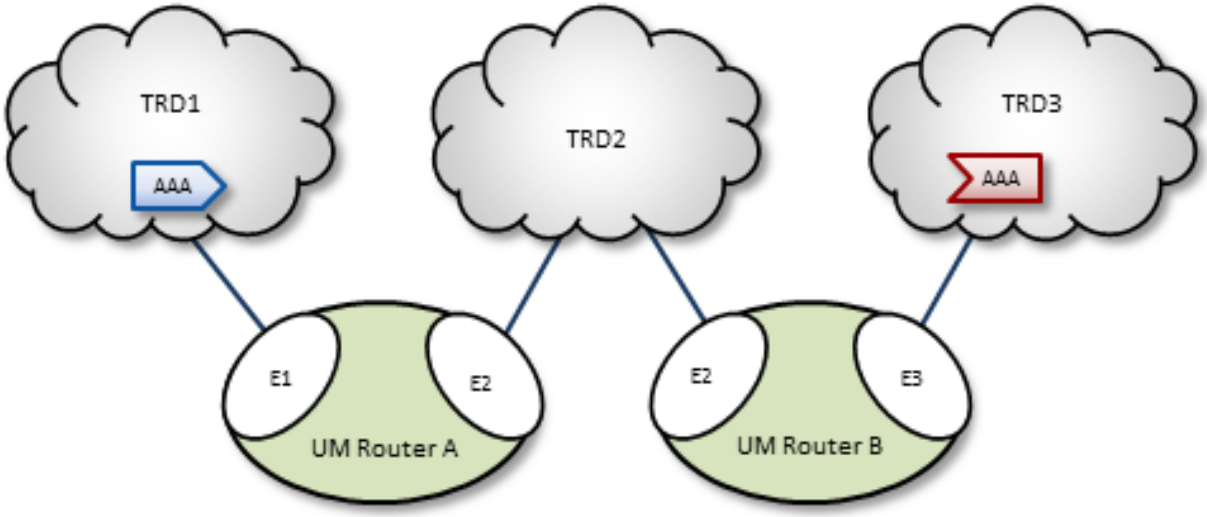
See configuration options: **transport_tcp_datagram_max_size (context)**, **transport_lbtrm_datagram_max_size (context)**, **transport_lbtru_datagram_max_size (context)**, **transport_lbtipc_datagram_max_size (context)**, and **transport_lbtsmx_datagram_max_size (source)**.

Also see **Message Fragmentation and Reassembly**.

Final note: the **resolver_datagram_max_size (context)** option also needs to be made the same in all instances of UM, including DROs.

3.2 Multi-Hop Forwarding

UM can resolve topics across a span of multiple DROs. Consider a simple example DRO deployment, as shown in the following figure.



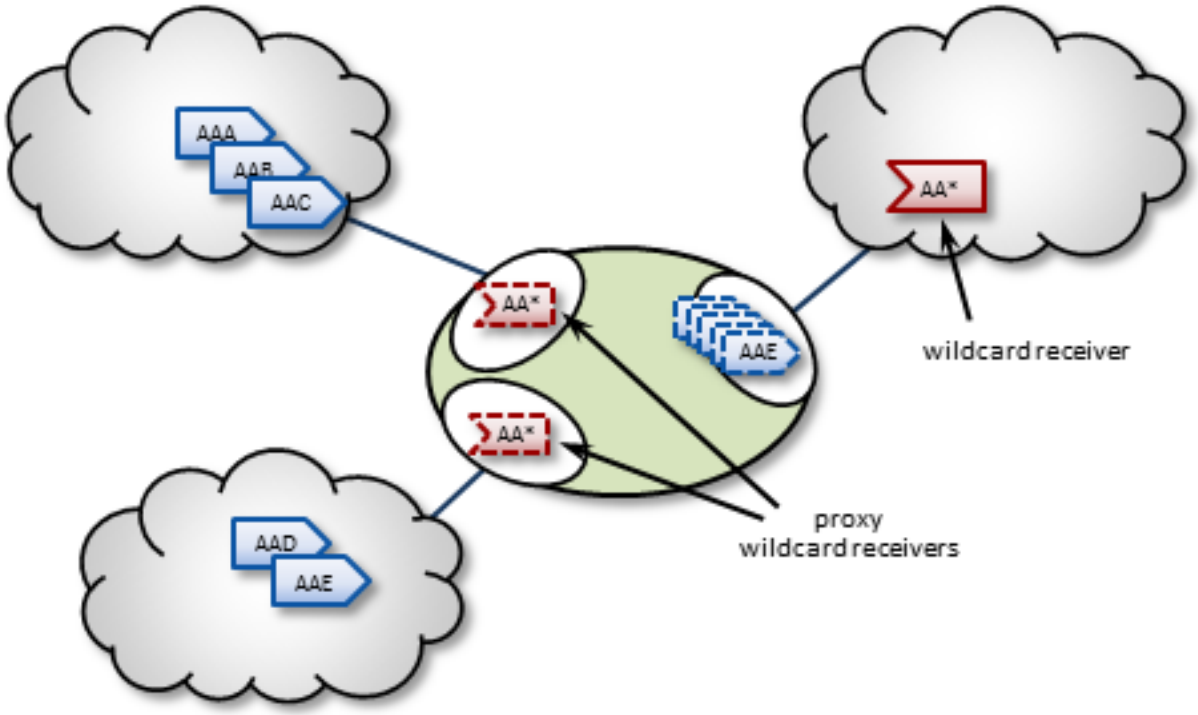
In this diagram, DRO A has two endpoint portals connected to topic resolution domains TRD1 and TRD2. DRO B also has two endpoint portals, which bridge TRD2 and TRD3. Endpoint portal names reflect the topic resolution domain to which they connect. For example, DRO A endpoint E2 interfaces TRD2.

TRD1 has a source for topic AAA, and TRD3, an AAA receiver. The following sequence of events enables the forwarding of topic messages from source AAA to receiver AAA.

1. Receiver AAA queries (issues a TQR).
2. DRO B, endpoint E3 (B-E3) receives the TQR and passes information about topic AAA to all other portals in the DRO. In this case, B-E2 is the only other portal.
3. In response, B-E2 creates a proxy receiver for AAA and sends a Topic Interest message for AAA into TRD2. The proxy receiver also issues a TQR, which in this case is ignored.
4. DRO A, endpoint E2 (A-E2) receives this Topic Interest message and passes information about topic AAA to all other portals in the DRO. In this case, A-E1 is the only other portal.
5. In response, A-E1 creates a proxy receiver for AAA and sends a Topic Interest message and TQR for AAA into TRD1.
6. Source AAA responds to the TQR by sending a TIR for topic AAA. In this case, the Topic Interest message is ignored.
7. The AAA proxy receiver created by A-E1 receives this TIR and requests that all DRO A portals with an interest in topic AAA create a proxy source for AAA.
8. In response, A-E2 creates a proxy source, which sends a TIR for topic AAA via TRD2.
9. The AAA proxy receiver at B-E2 receives this TIR and requests that all DRO B portals with an interest in topic AAA create a proxy source for AAA.
10. In response, B-E3 creates a proxy source, which sends a TIR for topic AAA via TRD3. The receiver in TRD3 joins the transport.
11. Topic AAA has now been resolved across both DROs, which forward all topic messages sent by source AAA to receiver AAA.

3.3 Routing Wildcard Receivers

The DRO supports topic resolution for wildcard receivers in a manner very similar to non-wildcard receivers. Wildcard receivers in a TRD issuing a WC-TQR cause corresponding proxy wildcard receivers to be created in portals, as shown in the following figure. The DRO creates a single proxy source for pattern match.



3.4 Forwarding Costs

Forwarding a message through a DRO incurs a cost in terms of latency, network bandwidth, and CPU utilization on the DRO machine (which may in turn affect the latency of other forwarded messages). Transiting multiple DROs adds even more cumulative latency to a message. Other DRO-related factors such as portal buffering, network bandwidth, switches, etc., can also add latency.

Factors other than latency contribute to the cost of forwarding a message. Consider a message that can be sent from one domain to its destination domain over one of two paths. A three-hop path over 1Gbps links may be faster than a single-hop path over a 100Mbps link. Further, it may be the case that the 100Mbps link is more expensive or less reliable.

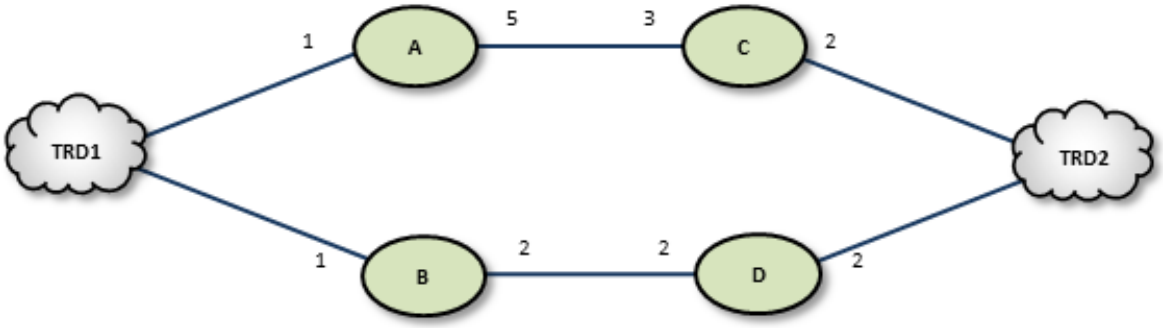
You assign forwarding cost values on a per-portal basis. When summed over a path, these values determine the cost of that entire path. A network of DROs uses forwarding cost as the criterion for determining the best path over which to resolve a topic.

3.5 DRO Routing

DROs have an awareness of other DROs in their network and how they are linked. Thus, they each maintain a topology map, which is periodically confirmed and updated. This map also includes forwarding cost information.

Using this information, the DROs can cooperate during topic resolution to determine the best (lowest cost) path over which to resolve a topic or to route control information. They do this by totaling the costs of all portals along each candidate route, then comparing the totals.

For example, the following figure shows two possible paths from TRD1 to TRD2: A-C (total route cost of 11) and B-D (total route cost of 7). In this case, the DROs select path B-D.



If a DRO or link along path B-D should fail, the DROs detect this and reroute over path A-C. Similarly, if an administrator revises cost values along path B-D to exceed a total of 12, the DROs reroute to A-C.

If the DROs find more than one path with the same lowest total cost value, i.e., a "tie", they select the path based on a node-ID selection algorithm. Since administrators do not have access to node IDs, this will appear to be a pseudo-random selection.

Note

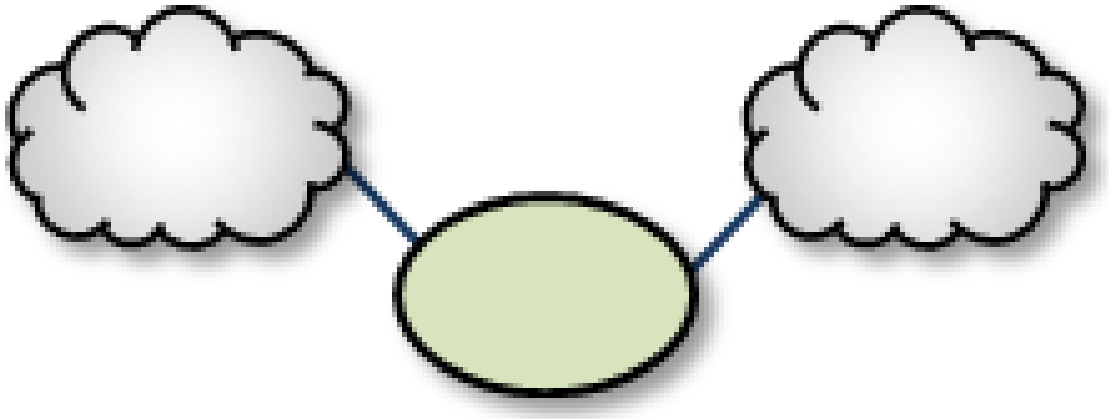
> You cannot configure parallel paths (such as for load balancing or Hot failover), as the DROs always select the lowest-cost path and only the lowest-cost path for all data between two points. However, you can devise an exception to this rule by configuring the destinations to be in different TRDs. For example, you can create an HFX Receiver bridging two receivers in different TRD contexts. The DROs route to both TRDs, and the HFX Receiver merges to a single stream for the application.

3.6 Routing Topologies

You can configure multiple DROs in a variety of topologies. Following are several examples.

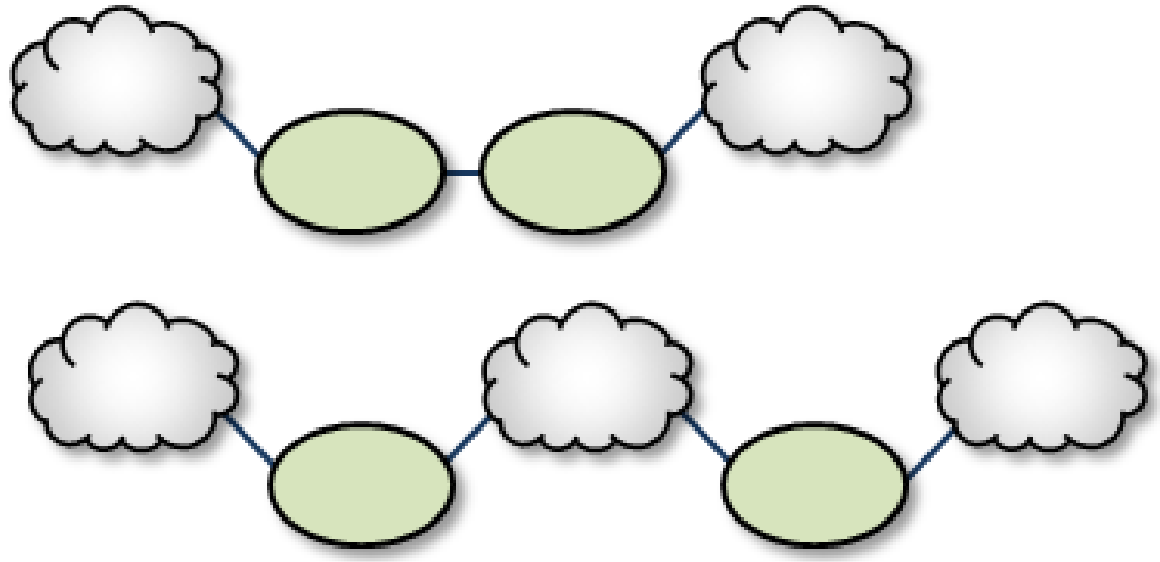
3.6.1 Direct Link

The Direct Link configuration uses a single DRO to directly connect two TRDs. For a configuration example, see Direct Link Configuration.



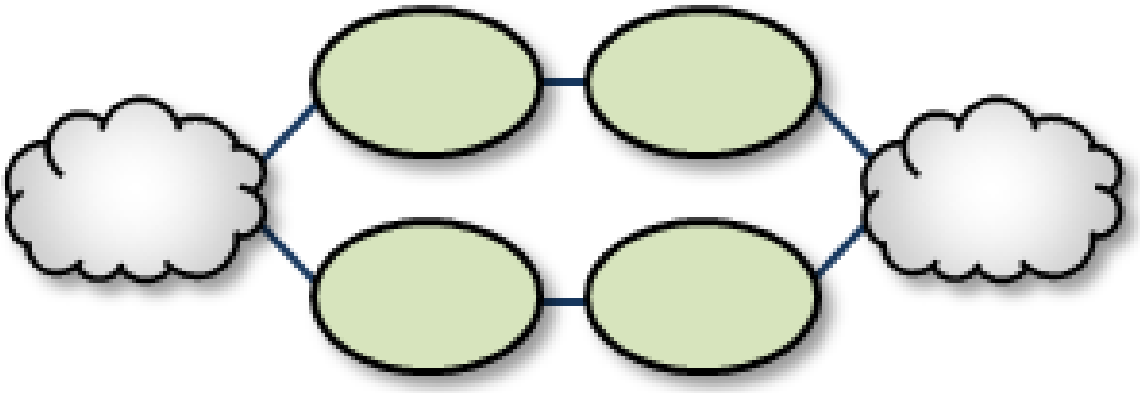
3.6.2 Single Link

A Single Link configuration connects two TRDs using a DRO on each end of an intermediate link. The intermediate link can be a "peer" link, or a transit TRD. For configuration examples, see Peer Link Configuration and Transit TRD Link Configuration.



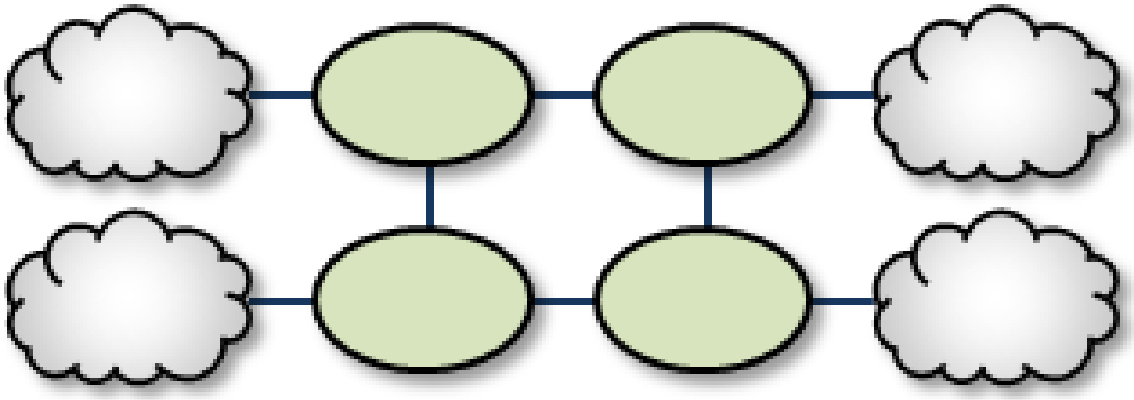
3.6.3 Parallel Links

Parallel Links offer multiple complete paths between two TRDs. However, UM will not load-balance messages across both links. Rather, parallel links are used for failover purposes. You can set preference between the links by setting the primary path for the lowest cost and standby paths at higher costs. For a configuration example, see Parallel Links Configuration.



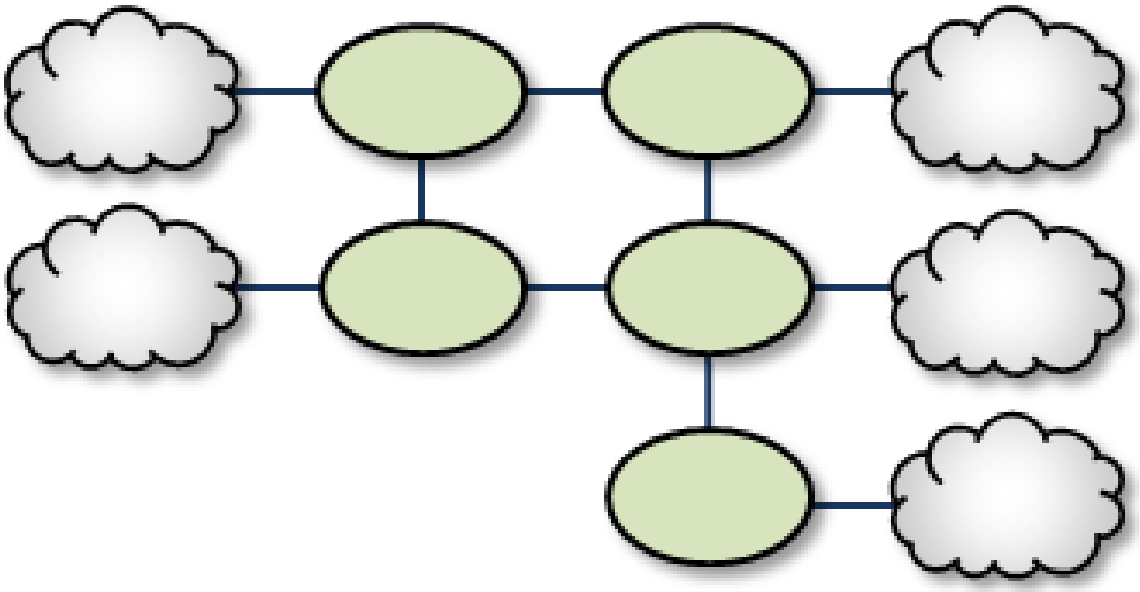
3.6.4 Loops

Loops let you route packets back to the originating DRO without reusing any paths. Also, if any peer-peer links are interrupted, the looped DROs are able to find an alternate route between any two TRDs.



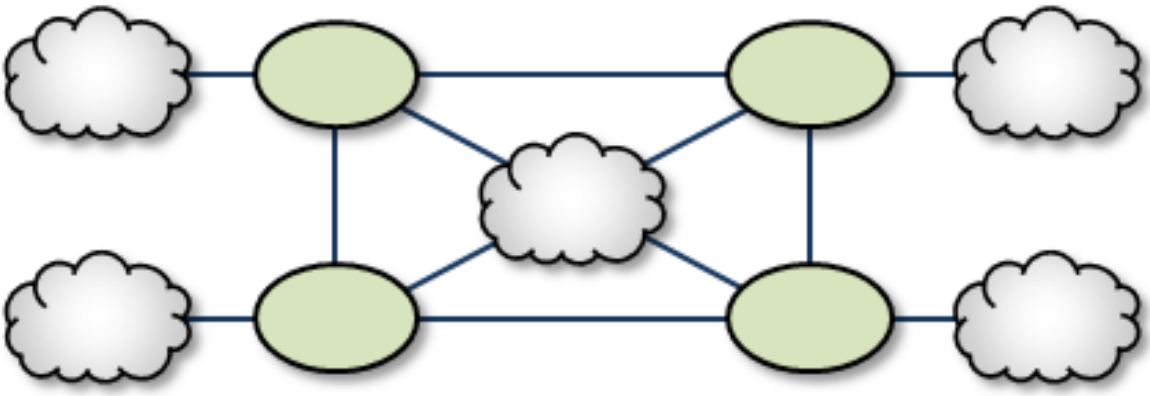
3.6.5 Loop and Spur

The Loop and Spur has a one or more DROs tangential to the loop and accessible only through a single DRO participating in the loop. For a configuration example, see Loop and Spur Configuration.



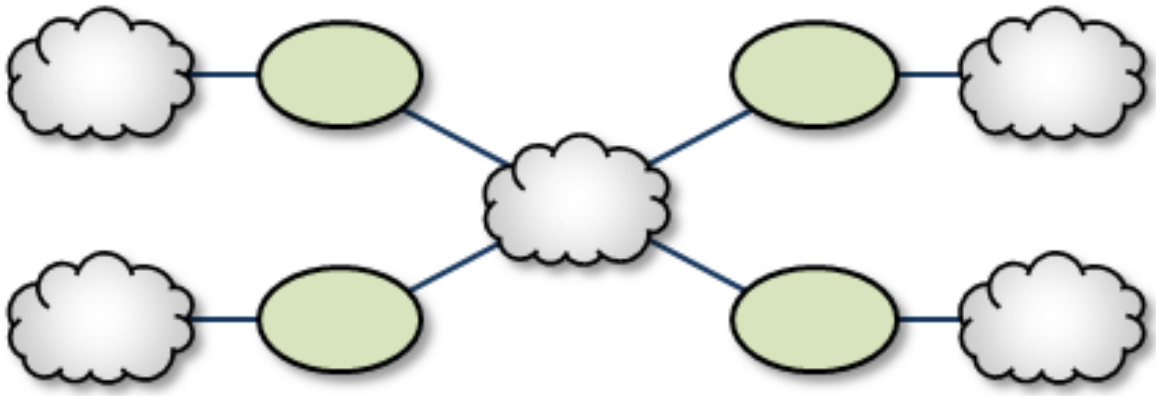
3.6.6 Loop with Centralized TRD

Adding a TRD to the center of a loop enhances its rerouting capabilities.



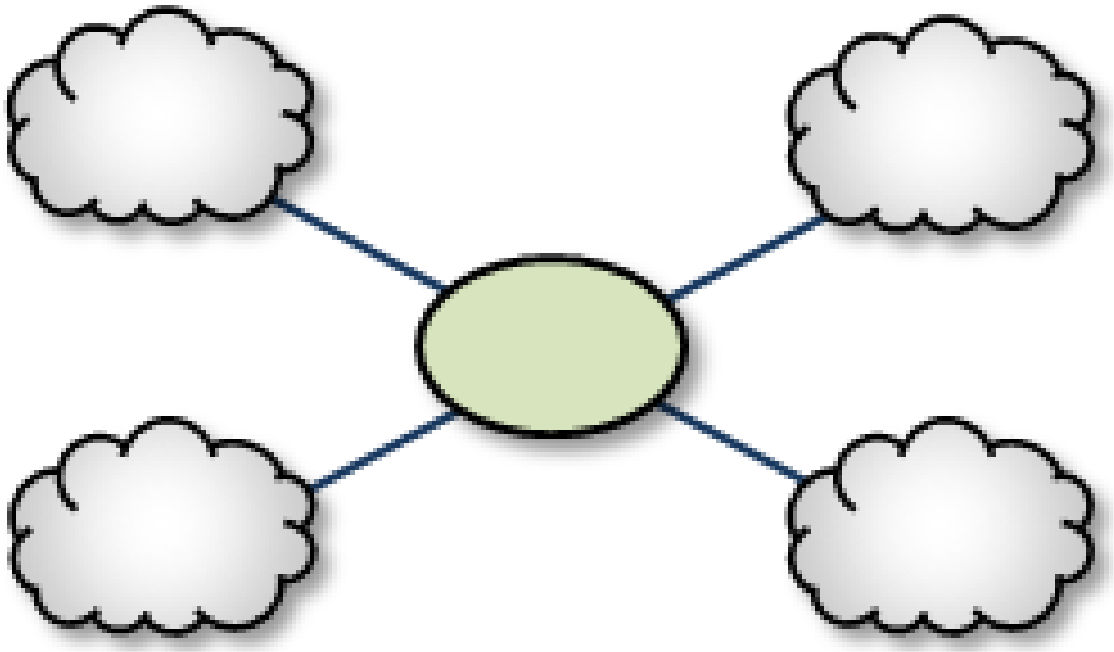
3.6.7 with centralized TRD

A Star with a centralized TRD does not offer rerouting capabilities but does provide an economical way to join multiple disparate TRDs.



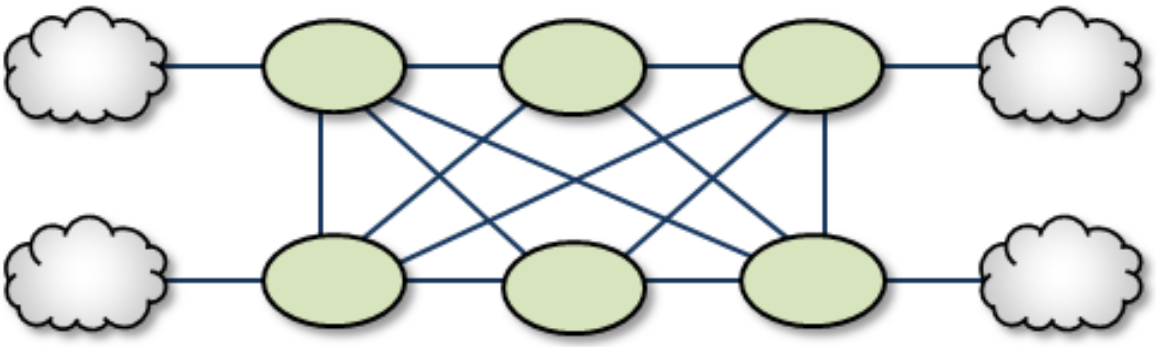
3.6.8 Star with Centralized DRO

The Star with a centralized DRO is the simplest way to bridge multiple TRDs. For a configuration example, see Star Configuration.



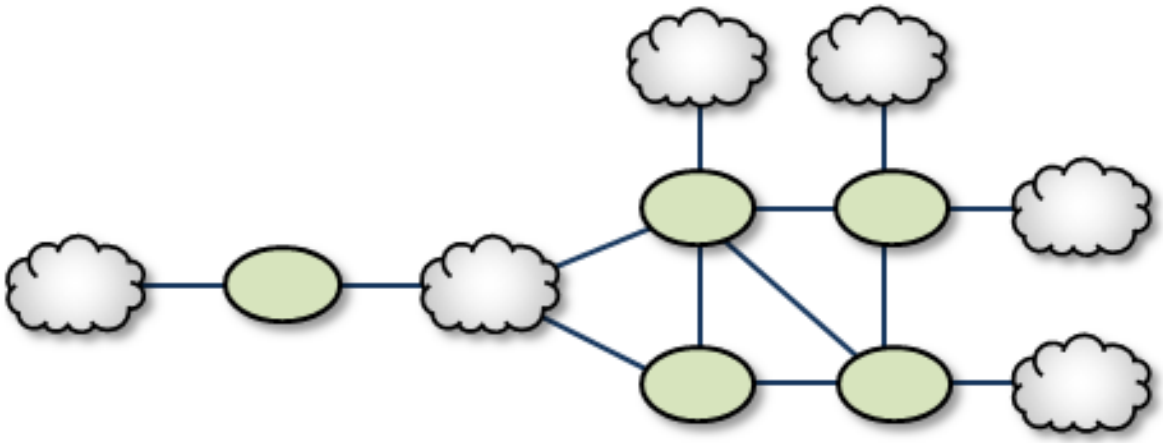
3.6.9 Mesh

The Mesh topology provides peer portal interconnects between many DROs, approaching an all-connected-to-all configuration. This provides multiple possible paths between any two TRDs in the mesh. Note that this diagram is illustrative of the ways the DROs may be interconnected, and not necessarily a practical or recommended application. For a configuration example, see Mesh Configuration.



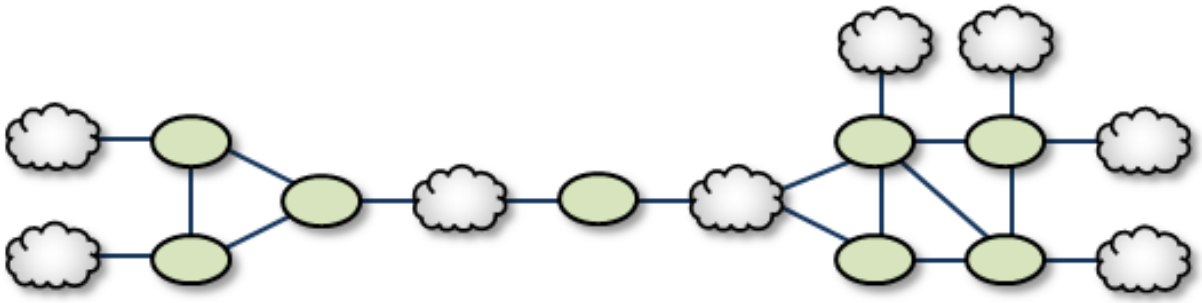
3.6.10 Palm Tree

The Palm Tree has a set of series-connected TRDs fanning out to a more richly meshed set of TRDs. This topology tends to pass more concentrated traffic over common links for part of its transit while supporting a loop, star, or mesh near its terminus.



3.6.11 Dumbbell

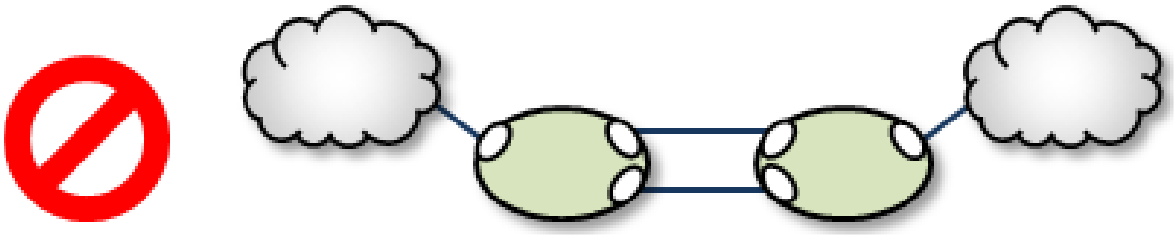
Similar to the Palm Tree, the Dumbbell has a funneled route with a loop, star, or mesh topology on each end.



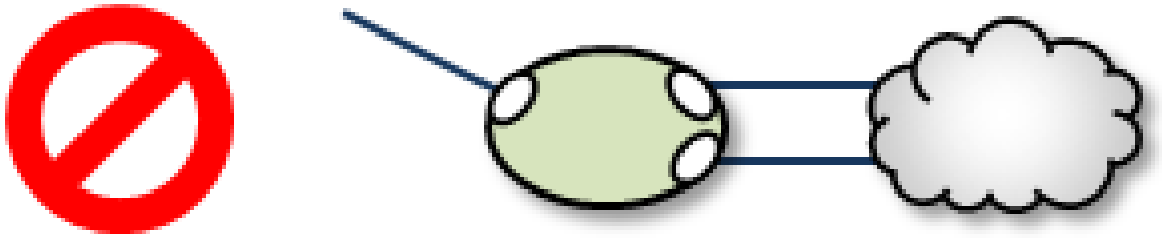
3.7 Unsupported Configurations

When designing DRO networks, do not use any of the following topology constructs.

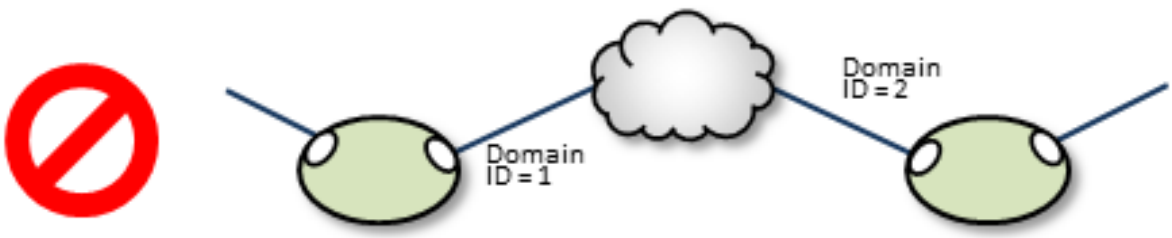
Two peer-to-peer connections between the same two DROs:



Two endpoint connections from the same DRO to the same TRD:



Assigning two different Domain ID values (from different DROs) to the same TRD:



3.8 UM Feature Compatibility

You must install the UM Dynamic Routing Option with its companion Ultra Messaging UMS, UMP, or UMQ product, and versions must match. While most UM features are compatible with the DRO, some are not. Following is a table of features and their compatibilities with the DRO.

UM Feature	DRO Compatible?	Notes
Connect and Disconnect Source Events	Yes, but see **Source Connect and Disconnect Events**	
Hot Failover (HF)	Yes	The DRO can pass messages sent by HF publishers to HF receivers, however the DRO itself cannot be configured to originate or terminate HF data streams.
Hot Failover Across Multiple Contexts	Yes	
Late Join	Yes	
Message Batching	Yes	
Monitoring/Statistics	Yes	
Multicast Immediate Messaging (MIM)	Yes	
Off-Transport Recovery (OTR)	Yes	
Ordered Delivery	Yes	
Pre-Defined Messages (PDM)	Yes	
Request/Response	Yes	
Self Describing Messaging (S↩ DM)	Yes	
Smart Sources	Partial	The DRO does not support proxy sources sending data via Smart Sources. The DRO does accept ingress traffic to proxy receivers sent by Smart Sources.
Source Side Filtering	Yes	The DRO supports transport source side filtering. You can activate this either at the originating TRD source, or at a downstream proxy source.
Source String	Yes, but see **Source Strings in a Routed Network**	
Transport Acceleration	Yes	
Transport LBT-IPC	Yes	
Transport LBT-RM	Yes	
Transport LBT-RU	Yes	
Transport LBT-SMX	Partial	The DRO does not support proxy sources sending data via LBT-S↩ MX. Any proxy sources configured for LBT-SMX will be converted to TCP, with a log message warning of the transport change. The D↩ RO does accept LBT-SMX ingress traffic to proxy receivers.

UM Feature	DRO Compatible?	Notes
Transport TCP	Yes	
Transport Services Provider (XSP)	No	
JMS, via UMQ broker	No	
Spectrum	Yes	The DRO supports UM Spectrum traffic, but you cannot implement Spectrum channels in DRO proxy sources or receivers.
UMP Implicit and **Explicit Acknowledgments**	Yes	
UMP Persistent Store	Yes	
UMP **Persistence Proxy Sources**	Yes	
UMP **Quorum/Consensus Store Failover**	Yes	
UMP **Managing RegIDs with Session IDs**	Yes	
UMP **RPP: Receiver-Paced Persistence** (RPP)	Yes	
UMQ **Brokered Queuing**	No	
UMQ **Ultra Load Balancing (ULB)**	No	
Ultra Messaging Desktop Services (UMDS)	Not for client connectivity to the UMDS server	
Ultra Messaging Manager (UMM)	Yes	Not for DRO management
UM SNMP Agent	No	
UMCache	No	
UM Wildcard Receivers	Yes	
Zero Object Delivery (ZOD)	Yes	

Chapter 4

DRO Implementation

4.1 DRO Configuration Overview

When the DRO daemon launches, it uses configuration option settings to determine its behavior and expectations. You specify option values in an XML configuration file, and reference the file from a command line argument.

Typically, you have a separate XML configuration file for each DRO, which contains structured configuration elements that describe aspects of the DRO. Within this XML configuration file, each endpoint portal definition points to a UM configuration file, which allow the portal to properly connect to its TRD.

4.2 Creating Applications for DRO Compatibility

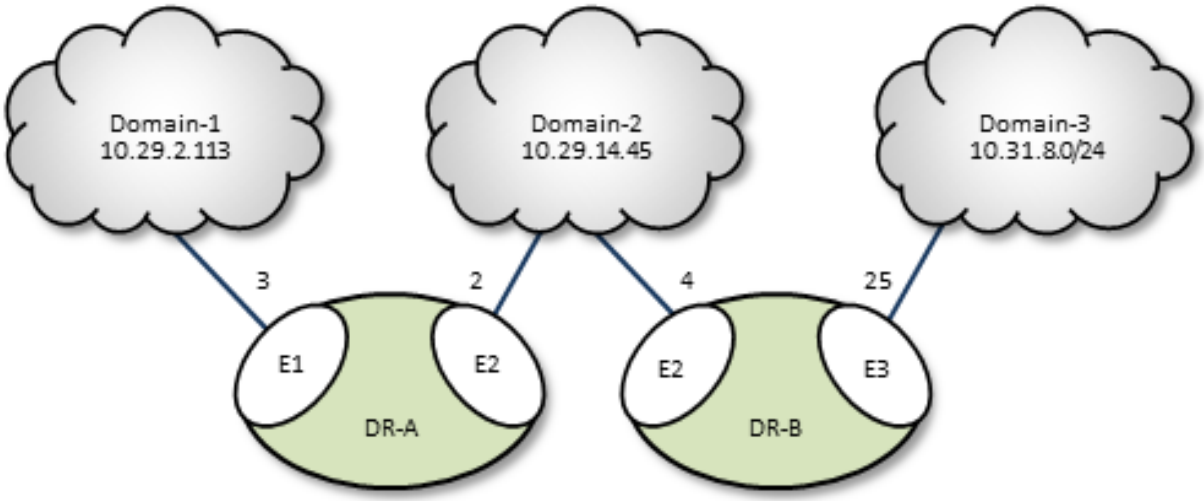
When developing messaging applications that use Ultra Messaging and, in particular, the DRO, please observe the following guidelines.

4.2.1 Naming and Identification

An important part to successfully implementing DROs is prudent and error-free naming of TRDs, DROs, portals, etc., as well as correct identification of IP addresses and ports. It is good practice to first design the DRO network by defining all connections and uniquely naming all DROs, portals, and TRDs. This works well as a diagram similar to some examples presented in this document. Include the following names and parameters in your design diagram:

- TRD names and IDs
- DRO names
- Portal names
- Portal costs

For example, a well-prepared DRO design could look like the following figure.



4.2.2 Portal Costs

A network of DROs uses forwarding cost as the criterion for determining the best (lowest cost) path over which to resolve a topic and route data. Forwarding cost is simply the sum of all portal costs along a multi-DRO path. Thus, total cost for the single path in the above example is 34. (Note that this is a non-real-world example, since costs are pointless without alternate routes to compare to.) You assign portal costs via the <cost> configuration option.

After the DRO network calculates its paths, if a new lower-cost source becomes available, receivers switch to that path.

4.2.3 Access Control Lists (ACL)

In the DRO, an Access Control List (ACL) is a method of blocking traffic from being forwarded from one TRD to another.

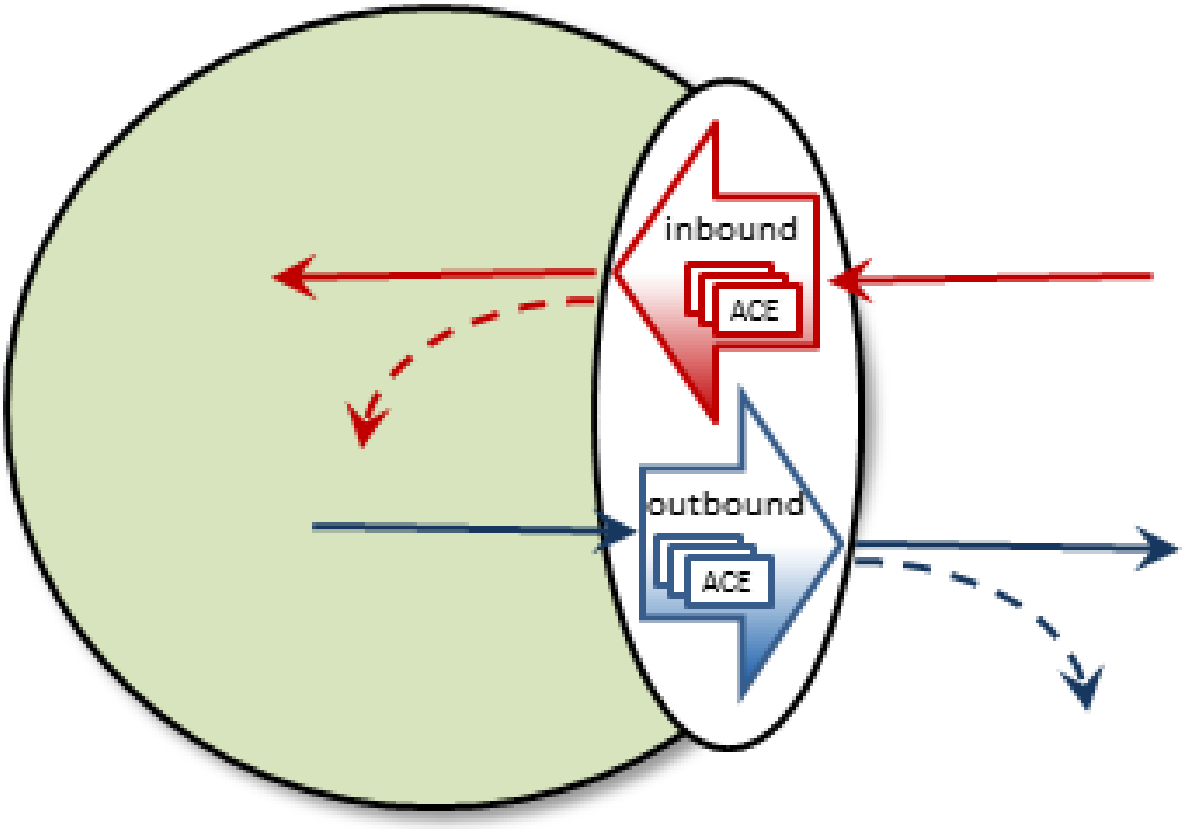
Typical applications for this feature are:

- Prevent unauthorized access to sensitive messages.
- Prevent overloading of bandwidth-limited WAN links, even in the face of accidental use of overly-permissive wildcard receivers.
- ACLs can be used to limit the amount of Topic Resolution traffic for topics on TRDs that don't need those topics. However, the use of wildcard receivers can result in TR traffic even for topics which are blocked from being forwarded.

You can apply Access Control Lists to one or more of a DRO's portals to filter traffic by topic, transport, transport session, etc. You configure an ACL in a DRO's XML configuration's <acl> element, as a child of an <endpoint> or <peer> portal. As messages are processed by the DRO, the portals use the ACLs to decide whether to reject the the messages or accept them.

Inbound vs. Outbound

There are two types of ACLs: inbound and outbound.



An inbound ACL tests messages from a source TRD on their way into a DRO portal, and decides whether to reject or accept them. If accepted, the messages can be forwarded to the appropriate destination portal(s).

An outbound ACL tests messages on their way out of a DRO portal, and decides whether to reject them, or transmit them to the destination TRD.

This distinction becomes especially important when a DRO has more than two portals. Messages rejected *inbound* cannot be forwarded at all. Messages rejected *outbound* can allow messages to be forwarded out some portals but not others.

An ACL contains one or more Access Control Entries (ACEs).

Access Control Entry (ACE)

An ACE specifies a set of message matching criteria, and an action to perform based on successful matches. The action is either accept (the message is made available for forwarding, based on interest) or reject (the message is dropped).

When more than one ACE is supplied in an ACL, messages are tested against each ACE in the order defined until a match is found, at which point the ACE specifies what to do (reject or accept).

An ACE contains one or more conditional elements.

Conditional Elements

Conditional elements do the work of testing various characteristics of messages to determine if they should be rejected or accepted (made available for forwarding).

When more than one conditional element is supplied in an ACE, received messages are tested against all of them to determine if the ACE should be applied.

There are two classes of conditional elements:

- Topic conditionals, which test the topic string of a message.

- Transport session conditionals, which test network transport session characteristics of a message.

Topic conditionals can be included on both inbound and outbound ACLs. The topic conditionals are:

- <topic> - tests for a specific topic name of messages,
- <pcre-pattern> - matches a group of topics according to a regular expression pattern,
- <regex-pattern> - deprecated, use <pcre-pattern> instead.

Transport session conditionals only apply to inbound ACLs (they are ignored for outbound). The transport session conditionals are:

- <transport> - tests the transport type of messages.
- <source-ip> - tests the IP address of the source or proxy source of messages.
- <multicast-group> - tests the destination multicast group of LBT-RM messages.
- <udp-destination-port> - tests the destination port of LBT-RM messages.
- <udp-source-port> - tests the source port of LBT-RM and LBT-RU messages.
- <tcp-source-port> - tests the source port of TCP messages.
- <xport-id> - tests the transport ID of LBT-IPC messages.

Conditional elements are children of the <ace> element. If you place multiple conditions within an ACE, the DRO performs an "and" operation with them. That is, all relevant conditions in the ACE must be true for the ACE to be applied to a message.

A portal will silently ignore conditional elements that don't apply. For example, if a transport conditional is used on an outbound ACL, or if a UDP-based conditional is present and a TCP message is received.

Reject by Default

An implicit "reject all" is at the end of every ACL, so the DRO rejects any topic not matched by any ACE. When an ACL is configured for a portal, rejection is the default behavior.

For example, to accept and forward only messages for topic ABC and reject all others:

```
<acl>
  <inbound>
    <ace match="accept">
      <topic>ABC</topic>
    </ace>
  </inbound>
</acl>
```

No "reject" ACE is needed since rejection is the default.

In contrast, to accept all messages *except* for topic ABC:

```
<acl>
  <inbound>
    <ace match="reject">
      <topic>ABC</topic>
    </ace>
    <ace match="accept">
      <topic>.*</topic>
    </ace>
  </inbound>
</acl>
```

The second ACE is used as a "match all", which effectively changes the default behavior to "accept".

ACE Ordering

Since the behavior for multiple ACEs is to test them in the order defined, ACEs should be ordered from specific to general.

In the example below, a user named "user1" is assigned to the LAN1 TRD. It is desired to forward all non-user-specific messages, but restrict user-specific message to only that user.

By ordering the ACEs as shown, messages for USER.user1 will be forwarded by the first ACE, but messages for USER.user2, etc. will be rejected due to the second ACE. Messages for topics not starting with "USER." will be forwarded by the third ACE.

```
<endpoint>
  <name>LAN1</name>
  <lbm-config>lan1.cfg</lbm-config>
  <domain-id>1</domain-id>
    <acl>
      <inbound>
        <ace match="accept">
          <topic>USER.user1</topic>
        </ace>
        <ace match="reject">
          <pcre-pattern>^USER\..*</pcre-pattern>
        </ace>
        <ace match="accept">
          <pcre-pattern>.*</pcre-pattern>
        </ace>
      </inbound>
    </acl>
</endpoint>
```

Note that the string in "<topic>USER.user1</topic>" is not a regular expression pattern, and therefore does not need any special escaping or meta characters. The "<pcre-pattern>^USER\..*</pcre-pattern>" is a regular expression, and therefore needs the "^" anchor and the "\." escape sequence.

4.2.4 Timers and Intervals

The DRO offers a wide choice of timer and interval options to fine tune its behavior and performance. There are interactions and dependencies between some of these, and if misconfigured, they may cause race or failure conditions.

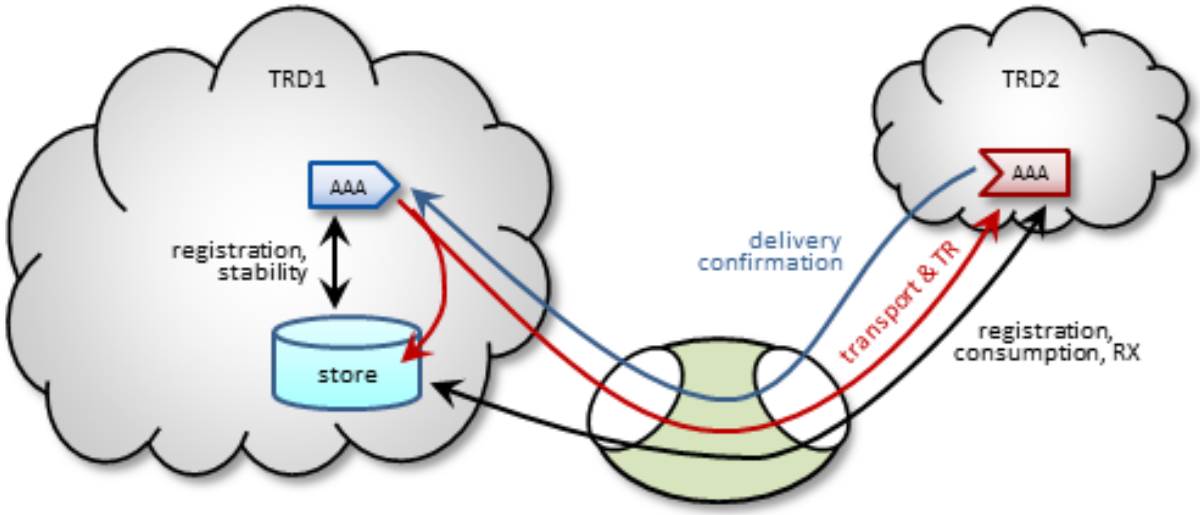
This manual's description of configuration options (see XML Configuration Reference), includes identification of such relationships. Please heed them.

4.2.5 Multicast Immediate Messaging Considerations

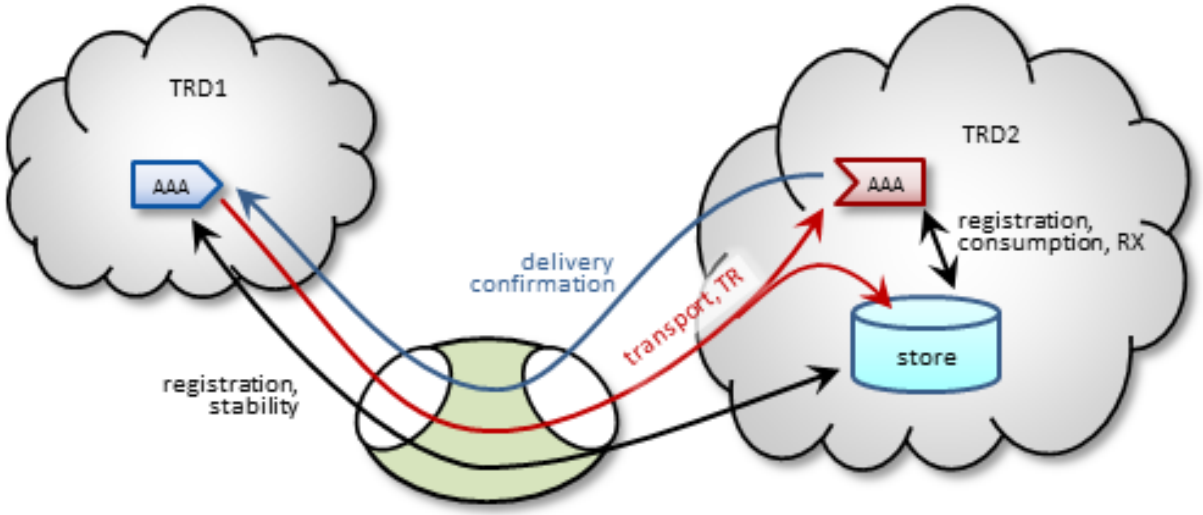
Multicast Immediate Messages (MIMs) may pass through the DRO. You cannot filter MIMs with Access Control Lists (ACL)-MIMs are forwarded to all TRDs. Informatica does not recommend using MIM for messaging traffic across the DRO. MIM is intended for short-lived topics and applications that cannot tolerate a delay between source creation and the sending of the first message. See also **Multicast Immediate Messaging**.

4.2.6 Persistence Over the DRO

The DRO supports UMP persistence by routing all necessary control and retransmission channels along with transport and topic resolution traffic. A typical implementation places the UMP persistent store in the same TRD as its registered source, as shown in the following figure.



The DRO also supports UMP implementations with the store located in a receiver's TRD, as shown in the following figure.



Note: For more reliable operation when using UMP with DROs, Informatica recommends enabling OTR.

4.2.7 Late Join and Off-Transport Recovery

The DRO supports sources and receivers configured for Late Join and/or Off-Transport Recovery (OTR). Retransmission requests and subsequent retransmissions are conducted across the entire path through the DRO network. A DRO's proxy sources do not have Late-Join/OTR retention buffers and hence, are not able to provide recovered messages.

4.2.8 Topic Resolution Reliability

Topic resolution can sometimes remain in a quiescent phase due to link interruption, preventing needed re-subscription topic resolution activity. Two ways you can address this are:

- For isolated incidents, call **lbm_context_topic_resolution_request()** (see example lbmtrreq.c). This restarts the sustaining phase.
- For more chronic problems, such as a DRO link (especially an endpoint link) over a WAN of questionable reliability, consider configuring Topic resolution to stay in the sustaining phase (options **resolver_advertisement_minimum_sustain_duration (source)** and **resolver_query_minimum_sustain_duration (receiver)**).

4.2.9 BOS and EOS Behavior Over the DRO

Through a network of DROs, a topic traverses a separate session for each link along its path. Thus, the DRO reports BOS/EOSs based on the activity between the proxy source transport and its associated receiver. There is no end-to-end, application-to-application reporting of the data path state. Also, in the case of multiple topics being assigned to multiple sessions, topics may find themselves with different session mates from hop to hop. Of course, this all influences when, and for which transport session, a topic's BOSs and EOSs are issued.

4.2.10 DRO Reliable Loss

The DRO can create a situation where a "reliable" transport (TCP or LBT-IPC) can experience out-of-order message delivery.

The DRO can perform a "protocol conversion" function. I.e. an originating source can use a UDP-based protocol (LBT-RM or LBT-RU), but the proxy source for a remote receiver can use a "reliable" protocol (TCP or LBT-IPC). With a UDP-based protocol, messages can arrive to the DRO network out of order, usually due to packet loss and recovery. However, when those out-of-order messages are forwarded across a "reliable" protocol (TCP or LBT-IPC), the receiver does not expect the sequence number gap, and immediately declares the out-of-order messages as unrecoverable loss. This, in spite of the fact that the missing message arrives shortly thereafter.

Starting in UM version 6.12, there are two new configuration options: **transport_tcp_dro_loss_recovery_timeout (receiver)** and **transport_lbtipc_dro_loss_recovery_timeout (receiver)**, which modify the receiver's behavior. Instead of declaring a gap immediately unrecoverable, a delay is introduced which is similar to what a UDP-based receiver uses to wait for lost and retransmitted datagrams. If the missing message arrives within the delay time, the messages are delivered to application without loss.

Be aware that this functionality is only used with "reliable" protocols published by a DRO's proxy source. If this delay feature is enabled, it will *not* apply to a "reliable" protocol that is received directly from the originating source.

Note however that you can get genuine gaps in the "reliable" data stream *without* recovery. For example, an overloaded DRO can drop messages. Or a DRO's proxy receiver can experience unrecoverable loss. In that case, the delay will have to expire before the missing messages are declared unrecoverable and subsequent data is delivered.

Attention

The delay times default to 0, which retains the pre-6.12 behavior of immediately declaring sequence number gaps unrecoverable. If you want this new behavior, you must configure the appropriate option.

4.3 Topology Configuration Examples

Following are example configurations for a variety of DRO topologies. These are the topology examples presented Routing Topologies.

In a real-world situation, you would have DRO XML configuration files with their portal interfaces referencing complete UM configuration files. However, for these examples, the referred domain configuration files are simplified to contain only information relevant to the applicable DRO. As part of this simplification, domain configuration files show interfaces for only one or two transport types.

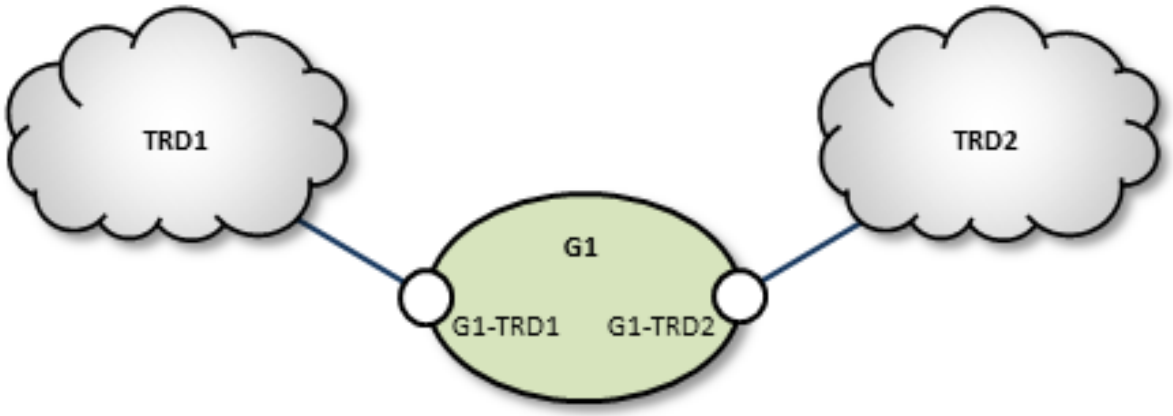
Also, IP addresses are provided in some cases and omitted in other cases. This is because initiator peer portals need to know the IP addresses (and port numbers) of their corresponding acceptor portals to establish connections, whereas endpoint portals communicate via topic resolution and thus, do not need to know IP addresses.

Note

Before designing any DRO implementations based on configurations or examples other than the types presented in this document, please contact your technical support representative.

4.3.1 Direct Link Configuration

This example uses a DRO to connect two topic resolution domain LANs.



TRD1 Configuration

This UM configuration file, trd1.cfg, describes TRD1 and is referenced in the DRO configuration file.

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

This DRO configuration file defines two endpoint portals. In the daemon section, we have turned on monitoring for the all endpoint portals in the DRO. The configuration specifies that all statistics be collected every 5 seconds and uses the lbm transport module to send statistics to your monitoring application, which runs in TRD1. See also UM Concepts, Monitoring UMS. The Web Monitor has also been turned on (port 15304) to monitor the performance of the DRO.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- G1 xml file- 2 endpoint portals -->
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <lbm-license-file>lic0014.txt</lbm-license-file>
    <monitor interval="5">
      <transport-module module="lbm" options="config=trd1.cfg"/>
    </monitor>
    <web-monitor>*:15304</web-monitor>
  </daemon>
  <portals>
    <endpoint>
      <name>G1-TRD1</name>
      <domain-id>1</domain-id>
      <lbm-config>trd1.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1-TRD2</name>
      <domain-id>2</domain-id>
      <lbm-config>trd2.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

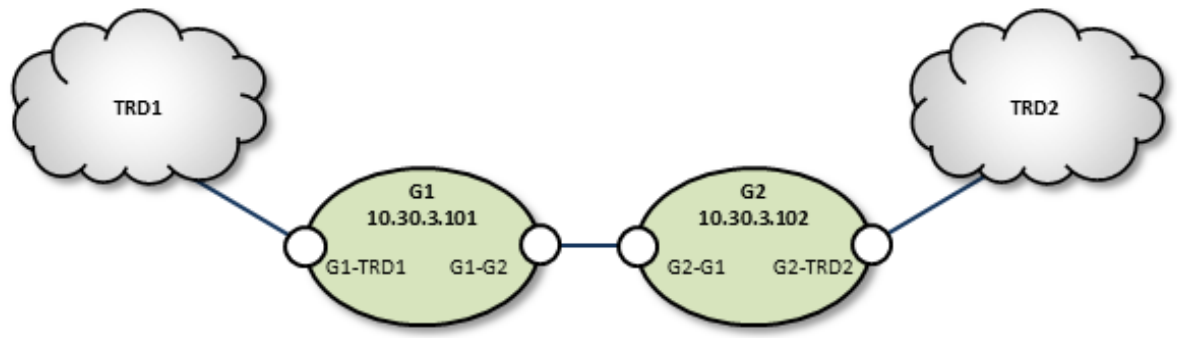
TRD2 Configuration

The configuration file trd2.cfg could look something like this.

```
# Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

4.3.2 Peer Link Configuration

In cases where the DRO connection between two TRDs must tunnel through a WAN or TCP/IP network, you can implement a DRO at each end, as shown in the example below.



TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

Following is an example of two companion peer portals (on different DROs) configured via DRO XML configuration file for a single TCP setup. Note that one must be an initiator and the other, an acceptor.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
       <log type="console"/>
    </daemon>
    <portals>
       <endpoint>
          <name>G1-TRD1</name>
          <domain-id>1</domain-id>
          <lbm-config>TRD1.cfg</lbm-config>
       </endpoint>
       <peer>
          <name>G1-G2</name>
          <single-tcp>
             <interface>10.30.3.100</interface>
             <initiator>
                <address>10.30.3.102</address>
                <port>26123</port>
             </initiator>
          </single-tcp>
       </peer>
    </portals>
</tnw-gateway>
```

G2 Configuration

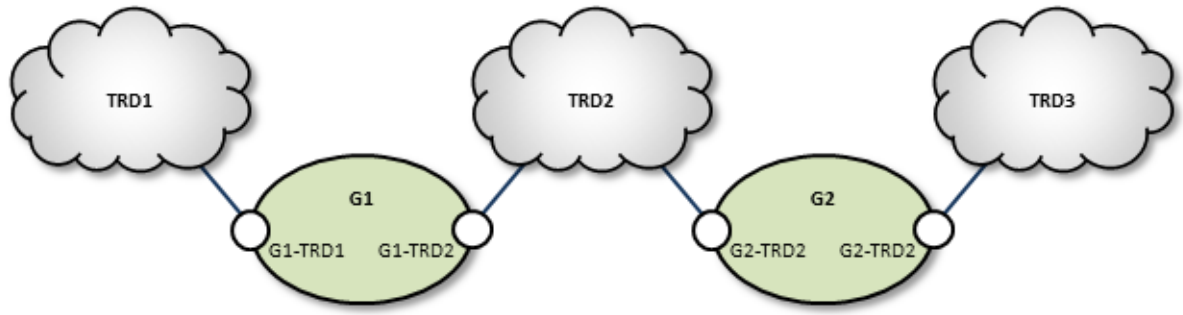
```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2-G1</name>
         <single-tcp>
           <interface>10.30.3.102</interface>
           <acceptor>
             <listen-port>26123</listen-port>
           </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G2-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## LAN2 Configuration Options ##
context request_tcp_interface 10.33.3.0/24
context resolver_multicast_port 13965
```

4.3.3 Transit TRD Link Configuration

This example, like the previous one, configures two localized DROs tunneling a connection between two TRDs, however, the DROs in this example are tunneling through an intermediate TRD. This has the added effect of connecting three TRDs.



TRD1 Configuration

```
## TRD1 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

G1 Configuration

Following is an example of two companion peer portals (on different DROs) configured via DRO XML configuration file for a single TCP setup. Note that one must be an initiator and the other, an acceptor.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G1-TRD1</name>
         <domain-id>1</domain-id>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G1-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## TRD2 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G2-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G2-TRD3</name>
         <domain-id>3</domain-id>
         <lbm-config>TRD3.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

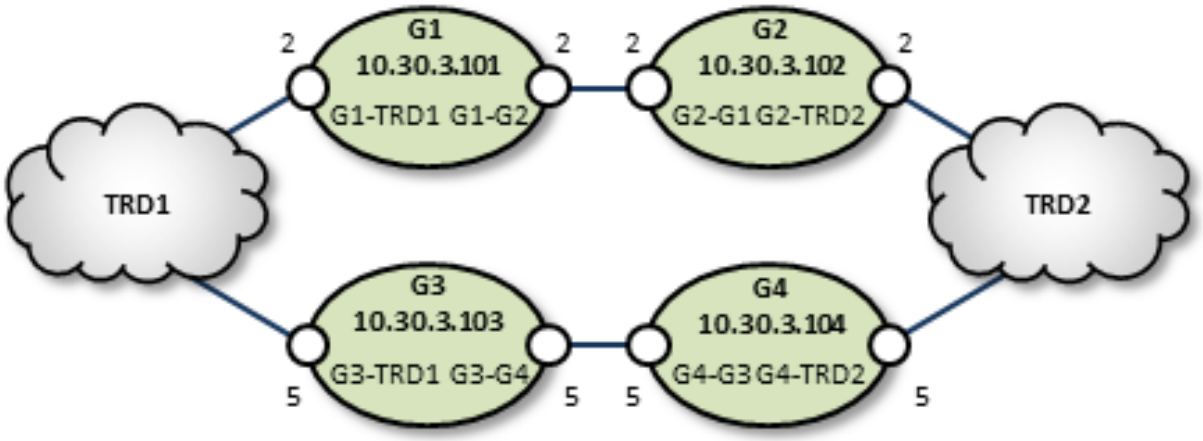
TRD3 Configuration

```
## TRD3 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

4.3.4 Parallel Links Configuration

This example is similar in purpose to the single link, peer-to-peer example, except that a second pair of DROs is added as a backup route. You can set one of these as a secondary route by assigning a higher cost to portals along the path. In this case we set G3 and G4's portal costs to 5, forcing the lower route to be selected only if the upper (G1, G2) route fails.

Also note that we have configured the peer portals for the leftmost or odd-numbered DROs as initiators, and the rightmost or even-numbered DRO peers as acceptors.



TRD1 Configuration

```
## TRD1 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G1-TRD1</name>
         <domain-id>1</domain-id>
         <cost>2</cost>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G1-G2</name>
         <cost>2</cost>
         <single-tcp>
            <interface>10.30.3.101</interface>
```

```
            <initiator>
               <address>10.30.3.102</address>
               <port>23745</port>
            </initiator>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2-G1</name>
         <cost>2</cost>
         <single-tcp>
            <interface>10.30.3.102</interface>
            <acceptor>
               <listen-port>23745</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G2-TRD2</name>
         <domain-id>2</domain-id>
         <cost>2</cost>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

G3 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G3-TRD1</name>
         <domain-id>1</domain-id>
         <cost>5</cost>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G3-G4</name>
         <cost>5</cost>
         <single-tcp>
            <interface>10.30.3.103</interface>
            <initiator>
               <address>10.30.3.104</address>
               <port>23746</port>
            </initiator>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

G4 Configuration

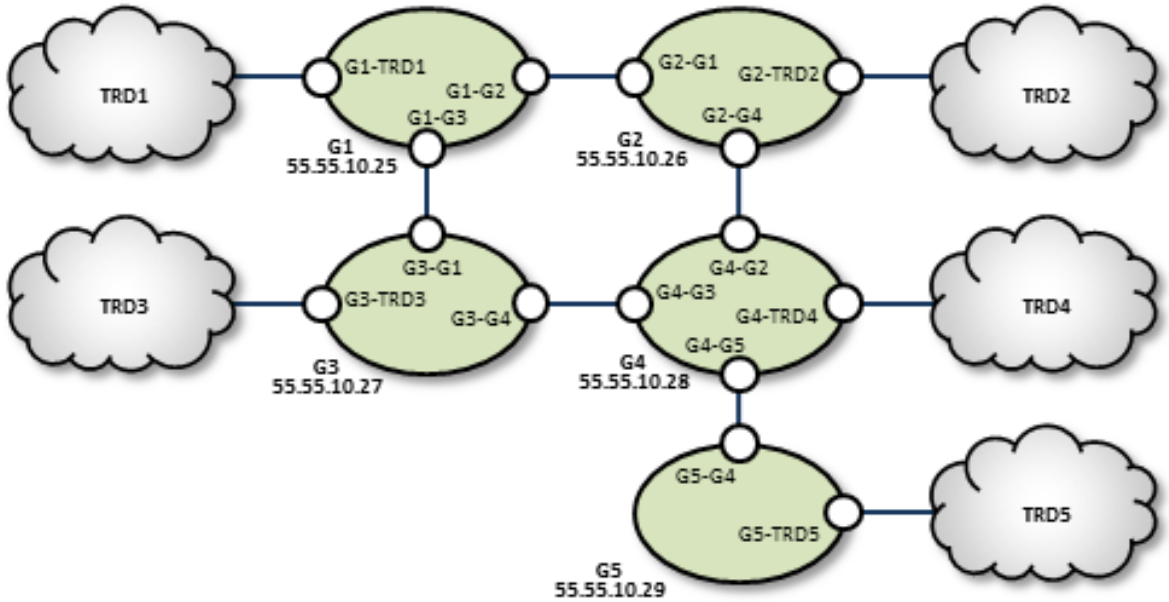
```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G4-G3</name>
         <cost>5</cost>
         <single-tcp>
            <interface>10.30.3.104</interface>
            <acceptor>
               <listen-port>23746</listen-port>
            </acceptor>
```

```
        </single-tcp>
      </peer>
      <endpoint>
         <name>G4-TRD2</name>
         <domain-id>2</domain-id>
         <cost>5</cost>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## TRD2 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

4.3.5 Loop and Spur Configuration



TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G1_to_G3</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.27</address>
```

```
            <port>23801</port>
         </initiator>
      </single-tcp>
    </peer>
    <peer>
       <name>G1_to_G2</name>
       <single-tcp>
          <initiator>
             <address>55.55.10.26</address>
             <port>23745</port>
          </initiator>
       </single-tcp>
    </peer>
    <endpoint>
       <name>G1_to_TRD1</name>
       <domain-id>1</domain-id>
       <lbm-config>TRD1.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2_to_G4</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.28</address>
               <port>23632</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G2_to_G1</name>
         <single-tcp>
            <acceptor>
               <listen-port>23745</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G2_to_TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

G3 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
```

```
      <peer>
         <name>G3_to_G4</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.28</address>
               <port>23754</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G3_to_G1</name>
         <single-tcp>
            <acceptor>
               <listen-port>23801</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G3_to_TRD3</name>
         <domain-id>3</domain-id>
         <lbm-config>TRD3.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

G4 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G4_to_G3</name>
         <single-tcp>
            <acceptor>
               <listen-port>23754</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G4_to_TRD4</name>
         <domain-id>4</domain-id>
         <lbm-config>TRD4.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G4_to_G2</name>
         <single-tcp>
            <acceptor>
               <listen-port>23632</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G4_to_G5</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.29</address>
               <port>23739</port>
            </initiator>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

TRD4 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.4.37.85
```

G5 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
```

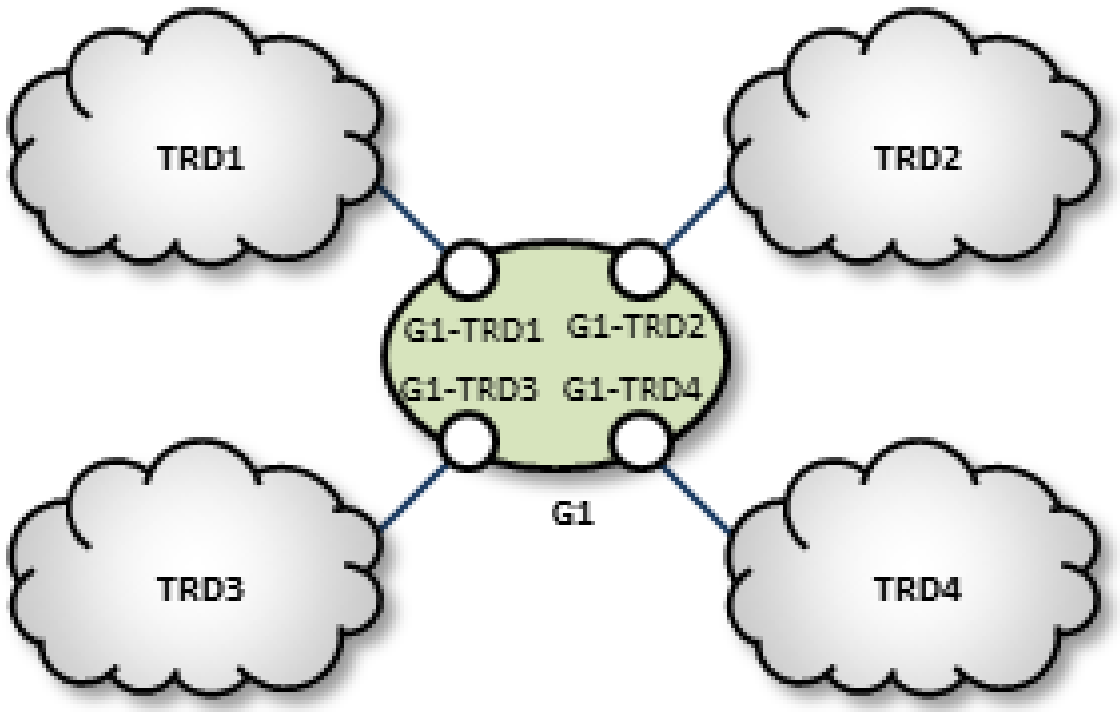
```
    </daemon>
    <portals>
      <endpoint>
         <name>G5_to_TRD5</name>
         <domain-id>5</domain-id>
         <lbm-config>TRD5.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G5_to_G4</name>
         <single-tcp>
            <acceptor>
               <listen-port>23739</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

TRD5 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.5.37.85
```

4.3.6 Star Configuration

This network consists of four TRDs. Within each TRD, full multicast connectivity exists. However, no multicast connectivity exists between the four TRDs.



G1 Configuration

The configuration for this DRO also has transport statistics monitoring and the WebMonitor turned on.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- UM GW xml file- 3 endpoint portals -->
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <lbm-license-file>lic0014.txt</lbm-license-file>
    <monitor interval="5">
      <transport-module module="lbm" options="config=trd1.cfg"/>
    </monitor>
    <web-monitor>*:15304</web-monitor>
  </daemon>
  <portals>
    <endpoint>
      <name>G1_to_TRD1</name>
      <domain-id>1</domain-id>
      <lbm-config>trd1.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD2</name>
      <domain-id>2</domain-id>
      <lbm-config>trd2.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD3</name>
      <domain-id>3</domain-id>
      <lbm-config>trd3.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD4</name>
      <domain-id>4</domain-id>
      <lbm-config>trd4.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

TRD2 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

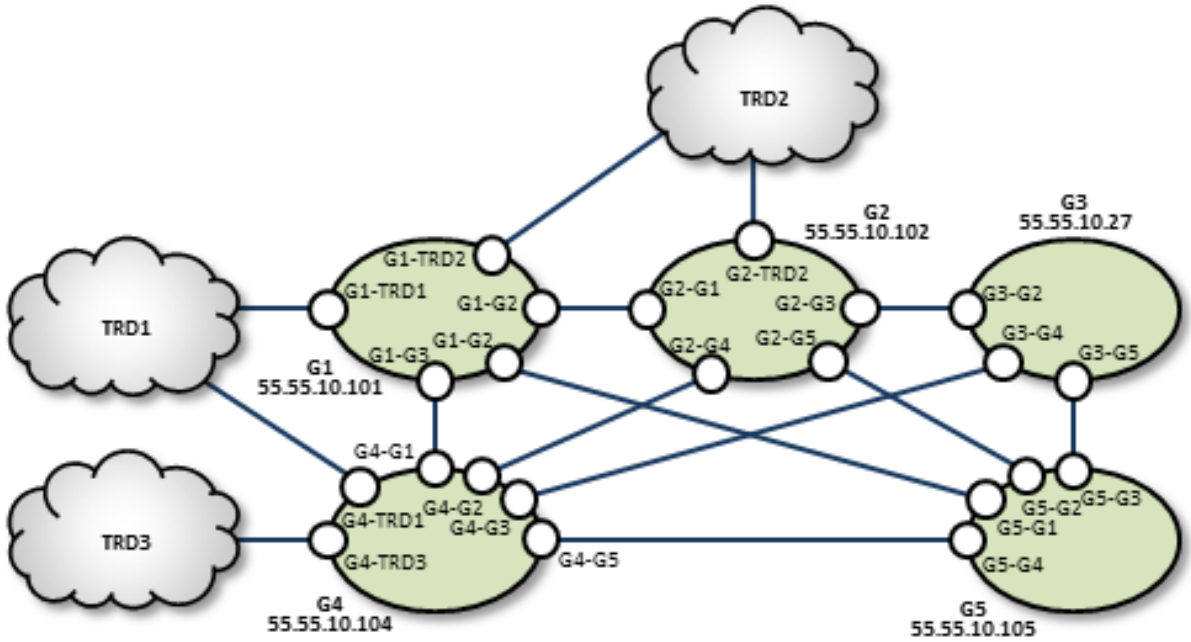
```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

TRD4 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.4.37.85
```

4.3.7 Mesh Configuration

The mesh topology utilizes many connections between many nodes, to provide a variety of alternate routes. However, meshes are not the best solution in many cases, as unneeded complexity can increase the chance for configuration errors or make it more difficult to trace problems.



TRD1 Configuration

```
### Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G1_to_G5</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.105</address>
               <port>23880</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G1_to_G4</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.104</address>
               <port>23801</port>
            </initiator>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G1_to_TRD1</name>
         <domain-id>1</domain-id>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G1_to_TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G1_to_G2</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.102</address>
```

```
            <port>23745</port>
         </initiator>
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2_to_G5</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.105</address>
               <port>23608</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G2_to_G4</name>
         <single-tcp>
            <acceptor>
               <listen-port>23831</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G2_to_G1</name>
         <single-tcp>
            <acceptor>
               <listen-port>23745</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G2_to_G3</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.103</address>
               <port>23632</port>
            </initiator>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G2_to_TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

G3 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G3_to_G5</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.105</address>
               <port>23739</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G3_to_G4</name>
         <single-tcp>
            <acceptor>
               <listen-port>23754</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
```

```
         <name>G3_to_G2</name>
         <single-tcp>
            <acceptor>
               <listen-port>23632</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

G4 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G4_to_G5</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.105</address>
               <port>23580</port>
            </initiator>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G4_to_TRD1</name>
         <domain-id>1</domain-id>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G4_to_TRD3</name>
         <domain-id>3</domain-id>
         <lbm-config>TRD3.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G4_to_G1</name>
         <single-tcp>
            <acceptor>
               <listen-port>23801</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G4_to_G3</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.103</address>
               <port>23754</port>
            </initiator>
         </single-tcp>
      </peer>
      <peer>
         <name>G4_to_G2</name>
         <single-tcp>
            <initiator>
               <address>55.55.10.102</address>
               <port>23831</port>
            </initiator>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

G5 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G5_to_G4</name>
         <single-tcp>
            <acceptor>
               <listen-port>23580</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G5_to_G1</name>
         <single-tcp>
            <acceptor>
               <listen-port>23880</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G5_to_G3</name>
         <single-tcp>
            <acceptor>
               <listen-port>23739</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
      <peer>
         <name>G5_to_G2</name>
         <single-tcp>
            <acceptor>
               <listen-port>23608</listen-port>
            </acceptor>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

4.4 Using UM Configuration Files with the DRO

Within the DRO configuration file, the endpoint portal's <lbm-config> element lets you import configurations from either a plain text or XML UM configuration file. However, using the XML type of UM configuration files provides the following advantages over plain text UM configuration files:

- You can apply UM attributes per topic and/or per context.
- You can apply attributes to all portals on a particular DRO using a UM XML template (instead of individual portal settings).
- Using UM XML templates to set options for individual portals lets the DRO process these settings in the <daemon> element instead of within each portal's configuration.

4.4.1 Setting Individual Endpoint Options

When setting endpoint options, first name the context of each endpoint in the DRO's XML configuration file.

```
<portals>
  <endpoint>
```

```
    <name>Endpoint_1</name>
    <domain-id>1</domain-id>
    <source-context-name>G1_E1</source-context-name>
    <lbm-attributes>
      <option name="request_tcp_interface" scope="context" value="10.29.4.0/24"/>
    </lbm-attributes>
  </endpoint>
  <endpoint>
    <name>G1-TRD2</name>
    <domain-id>2</domain-id>
    <receiver-context-name>G1_E2</source-context-name>
    <lbm-attributes>
      <option name="request_tcp_interface" scope="context" value="10.29.5.0/24" />
    </lbm-attributes>
  </endpoint>
</portals>
```

Then assign configuration templates to those contexts in the UM XML configuration file.

```
<application name="dro1" template="global">
  <contexts>
    <context name="G1_E1" template="G1-E1-options">
      <sources />
    </context>
    <context name="G1_E2" template="G1-E2-options">
      <sources />
    </context>
  </contexts>
</application>
```

You specify the unique options for each of this DRO's two endpoints in the UM XML configuration <templates> section used for G1-E1-options and G1-E2-options.

4.4.2 DRO and UM XML Configuration Use Cases

One advantage of using UM XML configuration files with the DRO is the ability to assign unique UM attributes to the topics and contexts used for the proxy sources and receivers (which plain text UM configuration files cannot do). The following example shows how to assign a different LBTRM multicast address to a source based on its topic.

Create a new UM XML configuration template for the desired topic name.

```
<template name="AAA-template">
  <options type="source">
    <option name="transport_lbtrm_multicast_address"
    default-value="225.2.37.88"/>
  </options>
</template>
```

Then include this template in the <application> element associated with the DRO.

```
<application name="dro1" template="global-options">
  <contexts>
    <context>
      <sources template="source-options">
        <topic topicname="AAA" template="AAA-template" />
      </sources>
    </context>
  </contexts>
</application>
```

It is also possible to assign UM attributes directly in the <application> tag. For example, the following specifies that a particular topic should use an LBT-RU transport.

```
<application name="dro1" template="dro1-common">
  <contexts>
    <context>
      <sources template="source-template">
        <topic topicname="LBTRU_TOPIC">
          <options type="source">
```

```
            <option name="transport" default-value="lbtru" />
            </options>
        </topic>
      </sources>
    </context>
  </contexts>
</application>
```

4.4.3 Sample Configuration

The following sample configuration incorporates many of the examples mentioned above. The DRO applies options to all UM objects created. The UM XML configuration file overwrites these options for two specific topics. The first topic, LBTRM_TOPIC, uses a different template to change its transport from TCP to LBTRM, and to set an additional property. The second topic, LBTRU_TOPIC, also changes its transport from TCP to a new value. However, its new attributes are applied directly in its associated topic tag, instead of referencing a template. In addition, this sample configuration assigns the rm-source template to all sources and receivers associated with the context endpt_1.

4.4.4 XML UM Configuration File

```
<?xml version="1.0" encoding="UTF-8" ?>
<um-configuration version="1.0">
  <templates>
    <template name="dro1-common">
      <options type="source">
        <option name="transport" default-value="tcp" />
      </options>
      <options type="context">
        <option name="request_tcp_interface" default-value="10.29.5.6" />
        <option name="transport_tcp_port_low" default-value="4400" />
        <option name="transport_tcp_port_high" default-value="4500" />
        <option name="resolver_multicast_address" default-value="225.2.37.88"/>
      </options>
    </template>
    <template name="rm-source">
      <options type="source">
        <option name="transport" default-value="lbtrm" />
        <option name="transport_lbtrm_multicast_address" default-value="225.2.37.89"/>
      </options>
    </template>
  </templates>
  <applications>
    <application name="dro1" template="dro1-common">
     <contexts>
        <context>
          <sources>
            <topic topicname="LBTRM_TOPIC" template="rm-source" />
            <topic topicname="LBTRU_TOPIC">
              <options type="source">
                <option name="transport" default-value="lbtru" />
                <option name="resolver_unicast_daemon" default-value="10.29.5.1:1234" />
              </options>
            </topic>
          </sources>
        </context>
        <context name="endpt_1">
          <sources template="rm-source"/>
        </context>
      </contexts>
    </application>
  </applications>
</um-configuration>
```

4.4.5 XML DRO Configuration File

This DRO uses the above XML UM configuration file, sample-config.xml, to set its UM options. It has three endpoints, one of which has the context endpt_1.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <xml-config>sample-config.xml</xml-config>
  </daemon>
  <portals>
    <endpoint>
      <name>Endpoint_1</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option name="context_name" scope="context" value="endpt_1" />
        <option name="request_tcp_interface" scope="context"
        value="10.29.4.0/24"/>
      </lbm-attributes>
    </endpoint>
    <endpoint>
      <name>Endpoint_2</name>
      <domain-id>2</domain-id>
      <lbm-attributes>
        <option name="request_tcp_interface" scope="context"
        value="10.29.5.0/24"/>
      </lbm-attributes>
    </endpoint>
    <endpoint>
      <name>Endpoint_3</name>
      <domain-id>3</domain-id>
      <lbm-attributes>
        <option name="request_tcp_interface" scope="context"
        value="10.29.6.0/24"/>
      </lbm-attributes>
    </endpoint>
  </portals>
</tnw-gateway>
```

4.5 Running the DRO Daemon

To run the DRO, ensure the following:

- Library environment variable paths are set correctly (LD_LIBRARY_PATH)
- The license environment variable LBM_LICENSE_FILENAME points to a valid DRO license file.
- The configuration file is error free.

Typically, you run the DRO with one configuration file argument, for example:

```
tnwgd gw1-config.xml
```

(FYI: "tnwgd" stands for "Twenty Nine West Gateway Daemon", a historical name for the DRO.)

The DRO logs version information on startup. The following is an example of this information:

```
Version 6.0 Build: Sep 26 2012, 00:31:33 (UMS 6.0 [UMP-6.0] [UMQ-6.0] [64-bit] Build: Sep 26 2012, 00:27:17
       ( DEBUG license LBT-RM LBT-RU LBT-IPC LBT-RDMA ) WC[PCRE 7.4 2007-09-21, regex, appcb] HRT[gettimeofday()])
```

Chapter 5

Man Pages for DRO

UM Message Routing services are provided by the DRO daemon (DRO).

There are two executables for the DRO, each with it's own man page:

- Tnwgd Man Page - Unix and Windows command-line interface.
- Tnwgds Man Page - Windows service interface.

(Note: "tnwg" stands for "Twenty Nine West Gateway", an older name for the DRO.)

5.1 Tnwgd Man Page

Unix and Windows command-line interface.

```
Purpose: UM Gateway daemon
Usage: tnwgd [options] configfile
Available options:
  -d, --dump-dtd            dump the configuration DTD to stdout
  -h, --help                display this help and exit
  -v, --validate            validate config-file then exit
  -f, --detach              detach from terminal
```

Description

The `tnwgd` command runs the DRO. It can be run interactively from a shell or command prompt, or from a script or batch file. (For use as a Windows Service, see Tnwgds Man Page.)

The **"configfile"** parameter is required and specifies the file path for the DRO's XML configuration file. See XML Configuration Reference for configuration details.

The **"-f"** option directs a Unix-based `tnwgd` to fork a child process which detaches from the controlling terminal. The `tnwgd` command normally remains attached to the controlling terminal and runs until interrupted. With "-f", the `tnwgd` command exits back to the shell, and the forked child continues running in the background.

The **"-d"** option dumps (prints) the DRO's XML DTD to standard output. After dumping the DTD, `tnwgd` exits. See DRO Configuration DTD for the DTD with comments removed.

The **"-v"** option validates the XML structure of the given configuration file against the DRO's XML DTD. After validating the configuration file's XML structure, `tnwgd` exits with status 0 for no errors, or non-zero if errors were found. For example:

```
tnwgd -v /um/dro_cfg.xml
```

Note that valid XML structure does not guarantee that the configuration file is completely correct. It must be tested on a running DRO.

The **"-h"** option prints the man page and exits.

Exit Status

The exit status from `tnwgd` is 0 for success and some non-zero value for failure.

5.2 Tnwgds Man Page

Windows service interface.

See **UM Daemons as Windows Services** for general information about UM daemons as Windows Services.

```
Purpose: UM Gateway daemon
Usage: tnwgds [options] [configfile]
Available options:
  -d, --dump-dtd             dump the configuration DTD to stdout
  -h, --help                 display this help and exit
  -v, --validate             validate config-file then exit
  -E, --env_var_file         update/set the environment Variable File
  -U, --unset-env-var-file   unset the environment variable file
  -s, --service=install      install the service passing configfile
  -s, --service=remove       delete/remove the service
  -s, --service=config       update configfile info to use configfile passed
  -e, --event-log-level      update/set service logging level.  This is the minimum logging
                             level to send to the Windows event log.  Valid values are:
                                NONE - Send no events
                                INFO
                                WARN - default
                                ERROR
```

Description

The `tnwgds` command has two functions:

- First, it lets the user supply Windows Service operating parameters, which the command saves into the Windows registry. Those operating parameters are subsequently used by the DRO Service. For details on setting Windows Service operating parameters, see **Configure the Windows Service**.
- Second, it provides Windows with the DRO executable to run as a Service.

The **"configfile"** parameter provides the file path for the DRO's XML configuration file. It is supplied in conjunction with the "-v" option or the "-s config" option (see below). See XML Configuration Reference for configuration details.

Note that valid XML structure does not guarantee that the configuration file is completely correct. It must be tested on a running DRO.

For **"-s install"** see **Install the Windows Service**.

For **"-s remove"** see **Remove the Windows Service**.

For **"-s config"**, **"-e"**, **"-E"**, and **"-U"**, see **Configure the Windows Service**.

The **"-d"** option dumps (prints) the DRO's XML DTD to standard output. After dumping the DTD, `tnwgds` exits.

The **"-v"** option validates the XML structure of the given configuration file against the DRO's XML DTD. After validating the configuration file's XML structure, `tnwgd` exits with status 0 for no errors, or non-zero if errors were found. For example:

```
tnwgds -v c:\um\dro_cfg.xml
```

Note that valid XML structure does not guarantee that the configuration file is completely correct. It must be tested on a running DRO.

The **"-h"** option prints the man page and exits.

Exit Status

The exit status from `tnwgd` is 0 for success and some non-zero value for failure.

Chapter 6

XML Configuration Reference

For controlling/configuring each DRO, you use a XML DRO configuration file, which also contains references to UM configuration files to extract needed information about the TRDs interfaced by endpoint portals.

An overview of the file format can be seen in the DRO Configuration DTD.

An XML DRO configuration file follows standard XML conventions. Element declarations or a pointer to a DTD file are not needed, as these are handled by the DRO.

6.1 File Structure

An XML DRO configuration file generally comprises two primary elements: <daemon> and <portals>. Organized and contained within these are option value assignments. <daemon> sub-containers let you set options global to the DRO. <portals> sub-containers let you configure each portal in the DRO individually.

In general, the order of the elements is important. Please refer to the examples and ensure proper element ordering.

XML DRO configuration files use the high-level structure shown in the following example. This example includes only some container elements, and only some options.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <uid>0</uid>
    <gid>0</gid>
    <pidfile>/path/file.pid</pidfile>
    <lbm-license-file>/path/file.lic</lbm-license-file>
    <topicmap/>
    <patternmap/>
    <monitor>
      <transport-module/>
      <format-module/>
    </monitor>
    <web-monitor>*:21000</web-monitor>
    <propagation-delay/>
    <xml-config>sample-config.xml</xml-config>
  </daemon>
  <portals>
    <endpoint>
      <name>Endpoint_1</name>
      <domain-id>1</domain-id>
      <cost>1</cost>
      <lbm-config>endpoint2.cfg</lbm-config>
      <lbm-attributes>
        <option name="context_name" scope="context" value="endpt_1" />
      </lbm-attributes>
      <acl>
        <inbound>
          <ace match="accept">
            <topic>ABC123</topic>
            <pcre-pattern >pattern</pcre-pattern >
```

```
            <regex-pattern >pattern</regex-pattern >
            <transport/>
            <source-ip/>
            <multicast-group/>
            <udp-source-port/>
            <udp-destination-port/>
            <tcp-source-port/>
            <xport-id/>
          </ace>
        </inbound>
        <outbound>
          <ace match="accept">
            <topic>ABC123</topic>
            <pcre-pattern >pattern</pcre-pattern >
            <regex-pattern >pattern</regex-pattern >
            <transport/>
            <source-ip/>
            <multicast-group/>
            <udp-source-port/>
            <udp-destination-port/>
            <tcp-source-port/>
            <xport-id/>
          </ace>
        </outbound>
      </acl>
    </endpoint>
    <peer>
      <name>Peer_1</name>
      <cost>1</cost>
      <single-tcp>
        <interface>
        <receive-buffer>
        <send-buffer>
        <keepalive>
        <nodelay>
        <initiator>
          <address>
          <port>
        </initiator>
        <acceptor>
          <listen-port>
        </acceptor>
      </single-tcp>
      <max-queue>
      <max-datagram>
      <batching>
        <min-length>
        <batch-interval>
      </batching>
      <lbm-config>peer1.cfg</lbm-config>
      <lbm-attributes>
        <option name="name" scope="scope" value="value" />
      </lbm-attributes>
      <acl> (see above)
      <topic-purge>
      <topic-interest-generate>
      <topic-domain-activity>
      <pattern-purge>
      <pattern-interest-generate>
      <pattern-domain-activity>
      <topic-use-check/>
      <pattern-use-check>
      <source-context-name>
      <receiver-context-name>
      <sqn-window>
      <context-query>
      <gateway-keepalive>
    </peer>
  </portals>
</tnw-gateway>
```

6.2 Elements Reference

6.2.1 Router Element "<tnw-gateway>"

Container for all options residing in the XML DRO configuration file. This is the top-level element.

- **Children:** <daemon>, <portals>

XML Attributes:

Attribute	Description	Valid Values	Default Value
version	The version of the DTD, which is currently. (This is not the product version.)	`"1.0"` - Current version of DTD.	(no default; must be specified)

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
  </daemon>
  <portals>
    ...
  </portals>
</tnw-gateway>
```

6.2.2 Router Element "<portals>"

Container for all endpoint and peer portal configuration information.

- **Parent:** <tnw-gateway>
- **Children:** <endpoint>, <peer>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
  </daemon>
  <portals>
    <endpoint>
      ...
    </endpoint>
    ...
  </portals>
</tnw-gateway>
```

6.2.3 Router Element "<peer>"

Container element for all configuration options of a single peer portal.

- **Parent:** <portals>

- **Children:** <name>, <cost>, <sourcemap>, <tcp>, <single-tcp>, <source-deletion-delay>, <max-queue>, <smart-batch>, <max-datagram>, <batching>, <lbm-config>, <lbm-attributes>, <acl>, <topic-purge>, <topic-interest-generate>, <topic-domain-activity>, <pattern-purge>, <pattern-interest-generate>, <pattern-domain-activity>, <topic-use-check>, <pattern-use-check>, <source-context-name>, <receiver-context-name>, <sqn-window>, <context-query>, <gateway-keepalive>, <publishing-interval>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
  </daemon>
  <portals>
    <peer>
      ...
    </peer>
    ...
  </portals>
</tnw-gateway>
```

6.2.4 Router Element "<publishing-interval>"

Configures the rate at which Daemon Statistics messages are published. See **Daemon Statistics** for general information on Daemon Statistics.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>, <daemon-monitor>
- **Children:** <group>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <daemon-monitor topic="umrouter.1">
      <lbm-config>/path/umrouter_monitor.cfg</lbm-config>
      <publishing-interval>
        <group name="default" ivl="5">
        <group name="gateway-config" ivl="30">
        <group name="portal-config" ivl="30">
      </publishing-interval>
      ...
    </daemon-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.5 Router Element "<group>"

Configures the rate at which one particular grouping of Daemon Statistics messages are published. See **Daemon Statistics** for general information on Daemon Statistics.

- **Parent:** <publishing-interval>

XML Attributes:

Attribute	Description	Valid Values	Default Value
name	Name of statistics group being configured.	**"default"** - Sets a default interval for all message types. **"gateway-config"** - Sets the interval for messages of type **tnwg_dstat_gatewaycfg_msg_t**. **"route-manager-topology"** - Sets the interval for messages of types **tnwg_rm_stat_grp_msg_t**. **"malloc-info"** - Sets the interval for messages of type **tnwg_dstat_mallinfo_msg_t**. **"portal-config"** - Sets the interval for messages of type **tnwg_pcfg_stat_grp_msg_t**. **"portal-stats"** - Sets the interval for messages of type **tnwg_dstat_portalstats_msg_t**	(no default; must be specified)
ivl	Time, in seconds, between publishing the statistics group being configured.	string	(no default; must be specified)

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <daemon-monitor topic="umrouter.1">
      <lbm-config>/path/umrouter_monitor.cfg</lbm-config>
      <publishing-interval>
        <group name="default" ivl="5">
        <group name="gateway-config" ivl="30">
        <group name="portal-config" ivl="30">
      </publishing-interval>
      ...
    </daemon-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.6 Router Element "<gateway-keepalive>"

Contains parameters for the keepalive signals sent from this peer portal. This is a DRO-level keepalive, not to be confused with the TCP-level <keepalive> element.

- **Cardinality:** 0 .. 1
- **Parent:** <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
idle	Determines if DRO keepalives should be sent only if no traffic has been sent or received in the last interval.	**"yes"** - Send only if no traffic has been exchanged. **"no"** - Send always, even of traffic has been exchanged.	**"yes"**
interval	Minimum interval, in milliseconds, between keepalive messages sent. Informatica recommends setting this to 2000 or greater. A value of 0 (zero) disables keepalives.	string	**"5000"**
timeout	Maximum time, in milliseconds, a peer can receive nothing from the companion before determining the connection is dead and disconnecting. We recommend setting this to 3 times the interval value.	string	**"15000"**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <gateway-keepalive idle="no" interval="1000"/>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.7 Router Element "<context-query>"

Determines timing characteristics for context name queries generated at this portal.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	Interval (in milliseconds) at which context queries are generated. Before changing the value of this option, please contact Informatica Support.	string	**"300000"**
max-contexts	Maximum number of contexts for which queries are generated at one time. Before changing the value of this option, please contact Informatica Support.	string	**"20"**
interval	Interval (in milliseconds) between groups of context queries. Before changing the value of this option, please contact Informatica Support.	string	**"200"**
timeout	Minimum time (in seconds) a context query must be unanswered before it is removed for the portal. Before changing the value of this option, please contact Informatica Support.	string	**"900"**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <context-query periodic-interval="25000" max-contexts="15" interval="180" timeout="875"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.8 Router Element "<sqn-window>"

Specifies the portal's awareness of received message sequence numbers, for the purpose of detecting duplicates.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
size	Determines the maximum number of topic (fragment) sequence numbers maintained in the window, for any given source. Must be a multiple of 8. Before changing the value of this option, please contact Informatica Support.	string	**"16384"**
increment	Determines the minimum increment, in topic (fragment) sequence numbers, by which the sequence number window is moved when the window size (below) is exceeded. Must be a multiple of 8, an even divisor of the window size, and less the window size. Before changing the value of this option, please contact Informatica Support.	string	**"2048"**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <sqn-window size="1024" increment="512"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.9 Router Element "<receiver-context-name>"

Specifies the portal receiver context name.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <receiver-context-name>RcvrContext01</source-context-name>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.10 Router Element "<source-context-name>"

Specifies the portal source context name.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <source-context-name>SourceContext01</source-context-name>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.11 Router Element "<pattern-use-check>"

Checks for interest in patterns at periodic intervals. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	The interval (in milliseconds) at which source pattern are checked to determine if there is no more interest. **This element is deprecated and has no function.**	string	"300000"

6.2.12 Router Element "<topic-use-check>"

Checks for interest in topics at periodic intervals. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	The interval (in milliseconds) at which source topics are checked to determine if there is no more interest. **This element is deprecated and has no function.**	string	"300000"

6.2.13 Router Element "<pattern-domain-activity>"

Determines how long a domain remains quiescent until it is determined inactive. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
timeout	Minimum time (in seconds) domain interest for a pattern must be refreshed before interest is removed for that domain. **This element is deprecated and has no function.**	string	"900"

6.2.14 Router Element "<pattern-interest-generate>"

Determines timing characteristics for interest message generation at this portal. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	Interval (in milliseconds) at which pattern interest is generated. **This element is deprecated and has no function.**	string	"300000"
max-patterns	Maximum patterns for which interest is generated at one time. **This element is deprecated and has no function.**	string	"300000"
interval	Interval (in milliseconds) between groups of patterns. **This element is deprecated and has no function.**	string	"200"

6.2.15 Router Element "<pattern-purge>"

Determines when this portal's proxy receivers can purge pattern. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	Interval (in milliseconds) at which receiver patterns are checked to determine if they can be purged. **This element is deprecated and has no function.**	string	"6000"

6.2.16 Router Element "<topic-domain-activity>"

Determines how long a domain remains quiescent until it is determined inactive. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
timeout	Minimum time (in seconds) domain interest for a topic must be refreshed before interest is removed for that domain. **This element is deprecated and has no function.**	string	"900"

6.2.17 Router Element "<topic-interest-generate>"

Determines timing characteristics for interest message generation at this portal. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	Interval (in milliseconds) at which topic interest is generated. **This element is deprecated and has no function.**	string	"300000"
max-topics	Maximum topics for which interest is generated at one time. **This element is deprecated and has no function.**	string	"20"
interval	Interval (in milliseconds) between groups of topics. **This element is deprecated and has no function.**	string	"200"

6.2.18 Router Element "<topic-purge>"

Determines when this portal's proxy receivers can purge topics. **This element is deprecated and has no function.**

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	Interval (in milliseconds) at which receiver topics are checked to determine if they can be purged. **This element is deprecated and has no function.**	string	"6000"

6.2.19 Router Element "<acl>"

Contains elements (inbound and outbound ACEs) that specify how an ACL (Access Control List) filters messages.

See Access Control Lists (ACL) for information on how ACLs work.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>
- **Children:** <inbound>, <outbound>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <acl>
        <inbound>
          <ace>
            <topic>AAA</topic>
          </ace>
        </inbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.20 Router Element "<outbound>"

Container for ACE elements, to separate outbound ACEs from inbound ACEs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Cardinality:** 0 .. 1
- **Parent:** <acl>
- **Children:** <ace>

Example:

Only forward messages for topics AAA and ABA.

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <acl>
        <outbound>
          <ace>
            <pcre-pattern>^A[AB]A$</pcre-pattern>
          </ace>
        </outbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.21 Router Element "<ace>"

Within an inbound or outbound ACL, you can have one or more "<ace>" elements. Each ACE (Access Control Entry) lets you match and accept or reject messages based on access control conditional elements, which are the elements contained within an "<ace>" element.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <inbound>, <outbound>
- **Children:** <topic>, <pcre-pattern>, <regex-pattern>, <transport>, <source-ip>, <multicast-group>, <udp-source-port>, <udp-destination-port>, <tcp-source-port>, <xport-id>

XML Attributes:

Attribute	Description	Valid Values	Default Value
match	This required attribute determines what to do with matched messages.	**"accept"** - Pass the message. **"reject"** - Block the message.	(no default; must be specified)

Example:

```
<tnw-gateway version="1.0">
  <portals>
    <endpoint>
      <name>LAN1</name>
      <lbm-config>lan1.cfg</lbm-config>
      <domain-id>1</domain-id>
        <acl>
          <inbound>
            <ace match="accept">
              <topic>ABC</topic>
            </ace>
            <ace match="accept">
              <topic>DEF</topic>
              <transport value=lbt-rm comparison=eq/>
            </ace>
            <ace match="accept">
              <topic>GHI</topic>
            </ace>
          </inbound>
        </acl>
    </endpoint>
    ...
```

6.2.22 Router Element "<xport-id>"

Defines a condition used in an ACE. Specifically, this matches the message's transport ID number (see **transport_lbtipc_id (source)**). This applies only to LBT-IPC transports.

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <xport-id comparison="equal" value="1234"/>
</ace>
```

6.2.23 Router Element "<tcp-source-port>"

Defines a condition used in an ACE. Specifically, this matches the message's TCP source port number (see **transport_tcp_port (source)**). This applies only to TCP transports.

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <tcp-source-port comparison="equal" value="1234"/>
</ace>
```

6.2.24 Router Element "<udp-destination-port>"

Defines a condition used in an ACE. Specifically, this matches the message's UDP destination port number (see **transport_lbtrm_destination_port (source)**). This applies only to LBT-RM transports.

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <udp-destination-port comparison="equal" value="1234"/>
</ace>
```

6.2.25 Router Element "<udp-source-port>"

Defines a condition used in an ACE. Specifically, matches the message's UDP source port number (see **transport_lbtrm_source_port_low (context)** and **transport_lbtru_port (source)**). This applies only to LBT-RM and LBT-RU transports.

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <udp-source-port comparison="equal" value="1234"/>
</ace>
```

6.2.26 Router Element "<multicast-group>"

Defines a condition used in an ACE. Specifically, this matches the message's multicast group address (see **transport_lbtrm_multicast_address (source)**). This applies only to LBT-RM transports.

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <multicast-group comparison="equal" value="1234"/>
</ace>
```

6.2.27 Router Element "<source-ip>"

Defines a condition used in an ACE. Specifically, this matches the message's source IP address. This applies only to TCP, LBT-RM, and LBT-RU transports.

Note

> The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The xport ID number to be compared.	string	(no default; must be specified)

Attribute	Description	Valid Values	Default Value
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal. **"lt"** - Matches if less than. **"lessthan"** - Matches if less than. **"le"** - Matches if less than or equal to. **"lessthanequal"** - Matches if less than or equal to. **"gt"** - Matches if greater than. **"greaterthan"** - Matches if greater than. **"ge"** - Matches if greater than or equal to. **"greaterthanequal"** - Matches if greater than or equal to.	(no default; must be specified)

Example:

```
<ace match="accept">
  <source-ip comparison="equal" value="1234"/>
</ace>
```

6.2.28 Router Element "<transport>"

Defines a condition used in an ACE. Specifically, this matches a UM transport type (see **transport (source)**).

Note

The message's originating source might be remote (i.e. not be in this DRO portal's TRD). In that case, this condition matches the TRD-local proxy source's characteristic, not the originating source.

This conditional element can only be used in inbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
value	The transport type to be matched.	**"tcp"** - TCP transport. **"lbt-rm"** - LBT-RM transport. **"lbtrm"** - LBT-RM transport. **"lbt-ru"** - LBT-RU transport. **"lbtru"** - LBT-RU transport. **"lbt-ipc"** - IPC transport. **"lbtipc"** - IPC transport.	(no default; must be specified)
comparison	Defines a match condition.	**"eq"** - Matches if equal. **"equal"** - Matches if equal. **"ne"** - Matches if not equal. **"notequal"** - Matches if not equal.	(no default; must be specified)

Example:

```
<ace match="accept">
  <transport comparison="equal" value="lbtrm"/>
</ace>
```

6.2.29 Router Element "<regex-pattern>"

Defines a condition used in an ACE. Specifically, this is a match pattern for one or more topics using a POSIX regular expression.

This element is deprecated. Please use <pcre-pattern> .

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

6.2.30 Router Element "<pcre-pattern>"

Defines a condition used in an ACE. Specifically, this is a match pattern for one or more topics using a Perl Compatible Regular Expression (PCRE).

This conditional element can be use in both inbound and outbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example 1:

This example will match patterns "ABC", "ABC789", and "ABC". It will not match "abc" or "123ABC".

```
<ace match="accept">
  <pcre-pattern>
    ^ABC.*
  </pcre-pattern>
</ace>
```

Example 2:

In this example, match any topic that has one or more spaces anywhere in the topic name. Note that the "xml↩ :space" attribute defaults to "default", which trims leading and trailing spaces. Therefore that attribute must set to "preserve", and the pattern must be combined onto a single line (to avoid newlines in the pattern):

```
<ace match="accept">
  <pcre-pattern xml:space="preserve"> </pcre-pattern>
</ace>
```

6.2.31 Router Element "<topic>"

Defines a condition used in an ACE. Specifically, this matches a topic name.

This conditional element can be use in both inbound and outbound ACLs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Parent:** <ace>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example 1:

Accept messages for topic "ABC":

```
<ace match="accept">
```

```
  <topic>ABC</topic>
</ace>
```

Example 2:

To match a topic name that includes a trailing space, you must use the change the xml:space attribute value:

```
<ace match="accept">
  <topic xml:space="preserve">ABC </topic>
</ace>
```

6.2.32 Router Element "<inbound>"

Container for ACE elements, to separate inbound ACEs from outbound ACEs.

See Access Control Lists (ACL) for information on how ACLs work.

- **Cardinality:** 0 .. 1
- **Parent:** <acl>
- **Children:** <ace>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <acl>
        <inbound>
          <ace>
            <topic>AAA</topic>
          </ace>
        </inbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.33 Router Element "<lbm-attributes>"

Container for individual UM-option-setting elements. It lets you set individual UM attributes without referencing a UM configuration file. These values override any values set via files referenced by <lbm-config>.

Note

Due to the order in which configuration options are processed, options specified in <lbm-attributes> do **not** override defaults set in <xml-config>. UM XML configuration files are flexible enough to allow proper overriding of common templates using named contexts. See <receiver-context-name> and <source-context-name>.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>
- **Children:** <option>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option scope="context" name="request_tcp_interface" value="10.28.5.5" />
        <option scope="context" name="response_tcp_interface" value="127.0.0.1" />
      </lbm-attributes>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.34 Router Element "<option>"

Lets you set an individual UM configuration option without referencing a UM configuration file. This value overrides any values set via files referenced by <lbm-config>.

Note

Some UM options specify interfaces, which can be done by supplying the device name of the interface. Special care must be taken when supplying device names. See **Interface Device Names and XML** for details.

- **Parent:** <lbm-attributes>

XML Attributes:

Attribute	Description	Valid Values	Default Value
scope	The type of object to which an option can apply.	**"receiver"** - Receiver option. **"context"** - Context option. **"source"** - Source option. **"wildcard_receiver"** - Wildcard Receiver option. **"event_queue"** - Event queue option.	(no default; must be specified)
name	The name of the option.	attr_name	(no default; must be specified)
value	The value for the option.	string	(no default; must be specified)

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option scope="context" name="request_tcp_interface" value="10.28.5.5" />
        <option scope="context" name="response_tcp_interface" value="127.0.0.1" />
      </lbm-attributes>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.35 Router Element "<lbm-config>"

Specifies the UM configuration file that contains configuration options associated with this portal.

Note that as of UM version 6.13, if one or more errors are discovered in the UM configuration file, the errors are written to the log file and the DRO continues running. I.e. errors in the UM configuration file are treated as warnings. See **Configuration Error Handling** for an explanation.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>, <daemon-monitor>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
      <lbm-config>/path/endpoint2.cfg</lbm-config>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.36 Router Element "<batching>"

Contains batching size and timing parameters for peer link implicit batching. This applies to data messages only: the DRO sends control messages immediately (flushing any batched data messages). Note: worst-case latency can be dramatically reduced by combining batching with <smart-batch>.

- **Cardinality:** 0 .. 1
- **Parent:** <peer>
- **Children:** <min-length>, <batch-interval>

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.37 Router Element "<batch-interval>"

Specifies the maximum interval (in milliseconds) between when the first message of a batch is queued until the batch is sent. A message stays in the batch queue until this value or <min-length> is met or exceeded (whichever occurs first). The minimum allowed value is 3 milliseconds.

If not specified, it defaults to 200 milliseconds.

- **Cardinality:** 0 .. 1
- **Parent:** <batching>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.38 Router Element "<min-length>"

Specifies the minimum length of a set of batched messages. When the total length of the batched messages reaches or exceeds this value, the batch is sent.

If not specified, it defaults to 8192 bytes.

- **Cardinality:** 0 .. 1
- **Parent:** <batching>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.39 Router Element "<max-datagram>"

Specifies the maximum size a peer portal will allow an outgoing datagram to be before fragmenting it.

If not specified, it defaults to 65535.

- **Cardinality:** 0 .. 1
- **Parent:** <peer>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <max-datagram>50000</max-datagram>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.40 Router Element "<smart-batch>"

Enables the smart batching algorithm used by the DRO when forwarding messages from one portal to another. Possible values are 0 (disable) and 1 (enable).

If not specified, it defaults to 0 (disabled).

In general, batching algorithms are used to increase throughput, but many such algorithms can produce latency outliers. The Smart Batching algorithm is designed to ensure low latencies by flushing the batching buffer when no more messages are waiting to be sent out the portal.

Smart batching works with both endpoint and peer portals. For endpoint portals, a UM configuration file may be provided to set the implicit_batching_minimum_length (source) option to a large value. For peer portals, the <batching> element may be used to set the <min-length> to a large value. In either case, large values are recommended and will not produce significant latency outliers.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <smart-batch>1</smart-batch>
      <batching>
        <min-length>4096</min-length>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.41 Router Element "<max-queue>"

Sets the maximum buffer size for blocking messages.

If not specified, this defaults to 1000000 bytes.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
  ...
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <max-queue>500000</max-queue>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.42 Router Element "<source-deletion-delay>"

Sets the time in milliseconds to wait after a source is detected as deleted before deleting the proxy source. Applies to both endpoint and peer portals.

Sources can be detected as being deleted by an EOS event at an endpoint portal, or by a route map change. Note that a route map change could be due to failure of a DRO or link within a network.

If not specified, source-deletion-delay defaults to 1000 milliseconds.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
  ...
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <source-deletion-delay>2000</source-deletion-delay>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.43 Router Element "<single-tcp>"

Contains elements for a peer portal's tcp settings, when configuring the peer for single-tcp operation.

- **Parent:** <peer>
- **Children:** <interface>, <receive-buffer>, <send-buffer>, <keepalive>, <nodelay>, <compression>, <tls>, <initiator>, <acceptor>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <interface>10.28.5.5/24</interface>
        <acceptor>
          <listen-port>23746</listen-port>
        </acceptor>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.44 Router Element "<acceptor>"

Contains the listen port address of the corresponding acceptor peer portal on another DRO, to which this peer is connected. This element is used in single-tcp peer configurations.

- **Parent:** <single-tcp>
- **Children:** <listen-port>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <acceptor>
          <listen-port>25000</port>
        </acceptor>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.45 Router Element "<listen-port>"

Contains port number on which an acceptor peer portal listens for connections from the initiating peer portal. There is no default for the port number, the initiating peer portal configuration must specify this port as its initiator port.

- **Parent:** <tcp>, <acceptor>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <acceptor>
          <listen-port>46000</listen-port>
        </acceptor>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.46 Router Element "<initiator>"

Contains the IP address and the port of the corresponding acceptor peer portal on another DRO, to which this peer is connected. This element is used in single-tcp peer configurations.

- **Parent:** <single-tcp>
- **Children:** <address>, <port>

XML Attributes:

Attribute	Description	Valid Values	Default Value
reconnect-interval	The time interval, in milliseconds, to wait before reconnecting to the companion portal if this connection is interrupted.	string	"5000"

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <initiator>
          <address>10.28.3.91</address>
          <port>25000</port>
        </initiator>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.47 Router Element "<port>"

Contains the IP port of the acceptor peer portal on another DRO, to which this initiator peer is connected. (As of UM version 6.10, dual TCP (<tcp>) is no longer supported. Please use <single-tcp> instead.)

- **Parent:** <companion>, <initiator>

Example:

```
<tnw-gateway version="1.0">
```

```
  ...
  <portals>
    <peer>
      <single-tcp>
        <initiator>
          <address>10.28.3.91</address>
          <port>25000</port>
        </initiator>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.48 Router Element "<address>"

Contains the IP address of the acceptor peer portal on another DRO, to which this initiator peer is connected via "single TCP". (As of UM version 6.10, dual TCP (<tcp>) is no longer supported. Please use <single-tcp> instead.)

- **Parent:** <companion>, <initiator>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <initiator>
          <address>10.28.3.91</address>
          <port>25000</port>
        </initiator>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.49 Router Element "<tls>"

Contains elements to configure peer link encryption.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>
- **Children:** <certificate>, <certificate-key>, <certificate-key-password>, <trusted-certificates>, <cipher-suites>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
          <certificate>test.crt<certificate>
          <certificate-key>test.key<certificate-key>
          <certificate-key-password>
            CorrectHorseBatteryStaple
```

```
          </certificate-key-password>
          <trusted-certificates>peers.crt<trusted-certificates>
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.50 Router Element "<cipher-suites>"

Defines the list of one or more (comma separated) names of cipher suites that are acceptable to this context. The names are in OpenSSL format (the ones with dashes). If more than suite name one is supplied, they should be in descending order of preference. When a remote context negotiates encrypted TCP, the two sides must find a cipher suite in common, otherwise the connection will be canceled.

The default is highly secure and is recommended.

- **Cardinality:** 0 .. 1
- **Parent:** <tls>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
          <cipher-suites>TLS_DHE_RSA_WITH_AES_256_GCM_SHA384<cipher-suites>
          ...
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.51 Router Element "<trusted-certificates>"

Specifies the path to a file containing one or more OpenSSL-compatible PEM-formatted TLS client certificates and certificate authorities. If this element is not supplied, the default behavior is to use the system-level trusted certificates and certificate authorities (operating-system dependent). The TLS server uses these trusted certificates to verify the identity of connecting clients. If a client connects and presents a certificate which is not in the server's trusted certificates file, the connection will be canceled.

- **Cardinality:** 0 .. 1
- **Parent:** <tls>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
```

```
          <trusted-certificates>peers.crt<trusted-certificates>
          ...
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.52 Router Element "<certificate-key-password>"

Specifies the passphrase needed to decrypt the server private key file specified by <certificate-key>.

- **Cardinality:** 0 .. 1
- **Parent:** <tls>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
          <certificate-key-password>
            CorrectHorseBatteryStaple
          </certificate-key-password>
          ...
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.53 Router Element "<certificate-key>"

Specifies the path to a file containing the private key associated with the "server" certificate specified by <certificate>. Note that this private key must be protected from intruders. For that reason, when the certificate and private key files are generated, the private key file is typically encrypted with a passphrase. The passphrase is supplied using <certificate-key-password>.

- **Parent:** <tls>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
          <certificate-key>test.key<certificate-key>
          ...
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.54 Router Element "<certificate>"

Specifies the path to a file containing an OpenSSL-compatible PEM-formatted certificate that will be presented as the TLS server certificate when a TLS connection is established by a client.

- **Parent:** <tls>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <tls>
          <certificate>test.crt<certificate>
          ...
        </tls>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.55 Router Element "<compression>"

Enables compression and sets the desired data compression algorithm for the peer link. Currently, only LZ4 lossless data compression is supported.

If not specified, no compression is used.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <compression>LZ4</compression>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.56 Router Element "<nodelay>"

Enables setting the TCP_NODELAY socket option on the peer link. Setting TCP_NODELAY disables Nagle's algorithm, which somewhat decreases the efficiency and throughput of TCP, but decreases the latency of individual messages.

By default, TCP_NODELAY is not set (maximizes efficiency).

- **Cardinality:** 0 .. 1

- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <nodelay/>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.57 Router Element "<keepalive>"

When present, enables a TCP keepalive signal transmission, which is disabled by default.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <keepalive/>
      . . .
      . . .
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.58 Router Element "<send-buffer>"

Contains the size of the TCP send buffer.

If not specified, the DRO uses the system default size.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <send-buffer>128000</send-buffer>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.59 Router Element "<receive-buffer>"

Contains the size of the TCP receive buffer.

If not specified, the DRO uses the system default size.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <receive-buffer>128000</receive-buffer>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.60 Router Element "<interface>"

Contains the IP host or network address for this peer portal, specified in dotted-decimal or CIDR format.

- **Cardinality:** 0 .. 1
- **Parent:** <tcp>, <single-tcp>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <peer>
      <single-tcp>
        <interface>10.28.5.5/24</interface>
        ...
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

6.2.61 Router Element "<tcp>"

DEPRECATED AND ELIMINATED AS OF UM 6.10. DO NOT USE. Contains elements for a peer portal's "dual TCP" settings. (As of UM version 6.10, dual TCP (<tcp>) is no longer supported. Please use <single-tcp> instead.)

- **Parent:** <peer>

- **Children:** <interface>, <listen-port>, <receive-buffer>, <send-buffer>, <keepalive>, <nodelay>, <compression>, <tls>, <companion>

6.2.62 Router Element "<companion>"

DEPRECATED AND ELIMINATED AS OF UM 6.10. DO NOT USE. Contains the IP address and the port of the companion peer portal on another DRO, to which this peer is connected via "dual TCP". (As of UM version 6.10, dual TCP (<tcp>) is no longer supported. Please use <single-tcp> instead.)

- **Parent:** <tcp>
- **Children:** <address>, <port>

XML Attributes:

Attribute	Description	Valid Values	Default Value
reconnect-interval		string	

6.2.63 Router Element "<sourcemap>"

Sets the size of the peer portal's source map. This normally does not need to be modified, but if very large numbers of topics are being used, a larger value might improve efficiency.

- **Cardinality:** 0 .. 1
- **Parent:** <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
size	Number of entries in the source map. Value must be a power of 2 (e.g., 1024, 2048, ...).	string	`"131072"`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <sourcemap size="131072"/>
      . . .
      . . .
    </peer>
  </portals>
</tnw-gateway>
```

6.2.64 Router Element "<cost>"

Assigns a positive non-zero integer cost to the portal.

If not specified, it defaults to 1. See Forwarding Costs.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>, <peer>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
  ...
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>25</cost>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.65 Router Element "<name>"

Lets you set a name for this DRO (do not duplicate for any other known DROs), or for the name of an endpoint or peer portal. Each portal name must be unique within the DRO.

If not specified, no name is assigned.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>, <endpoint>, <peer>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**`"default"`** - Trim whitespace. **`"preserve"`** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <name>DRO1</name>
  </daemon>
  <portals>
    <endpoint>
      <name>endpoint1</name>
      ...
    </endpoint>
  </portals>
  ...
</tnw-gateway>
```

6.2.66 Router Element "<endpoint>"

Container element for all configuration options of a single endpoint portal.

- **Parent:** <portals>
- **Children:** <name>, <domain-id>, <cost>, <source-deletion-delay>, <max-queue>, <smart-batch>, <lbm-config>, <lbm-attributes>, <acl>, <topic-resolution>, <late-join>, <topic-purge>, <topic-interest-generate>, <topic-domain-activity>, <pattern-purge>, <pattern-interest-generate>, <pattern-domain-activity>, <remote-topic>, <remote-pattern>, <source-context-name>, <receiver-context-name>, <sqn-window>, <context-query>, <publishing-interval>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
  ...
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>1</cost>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.67 Router Element "<remote-pattern>"

Determines timings and limits for determination of continued pattern interest at this portal.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>

XML Attributes:

Attribute	Description	Valid Values	Default Value
check-interval	Interval (in milliseconds) between checking individual patterns for continued interest. Before changing the value of this option, please contact Informatica Support.	string	"90000"
max-patterns	Maximum number of patterns to check at a time. Before changing the value of this option, please contact Informatica Support.	string	"100"
timeout	Minimum time (in milliseconds) remote interest for a pattern must be refreshed before interest is removed for that domain. Before changing the value of this option, please contact Informatica Support.	string	"300000"

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <remote-pattern check-interval="80000" max-topics="80" timeout="250000"/>
    </endpoint>
```

```
  </portals>
</tnw-gateway>
```

6.2.68 Router Element "<remote-topic>"

Determines timings and limits for determination of continued topic interest at this portal.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>

XML Attributes:

Attribute	Description	Valid Values	Default Value
check-interval	Interval (in milliseconds) between checking individual topics for continued interest. Before changing the value of this option, please contact Informatica Support.	string	`"90000"`
max-topics	Maximum number of topics to check at a time. Before changing the value of this option, please contact Informatica Support.	string	`"100"`
timeout	Minimum time (in milliseconds) remote interest for a topic must be refreshed before interest is removed for that domain. Before changing the value of this option, please contact Informatica Support.	string	`"300000"`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <remote-topic check-interval="80000" max-topics="80" timeout="250000"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.69 Router Element "<late-join>"

DEPRECATED AND ELIMINATED. DO NOT USE. Determines how Late Join is handled by this endpoint portal.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>

XML Attributes:

Attribute	Description	Valid Values	Default Value
provide		`"source"` `"always"` `"never"`	
forward		`"yes"`	
		`"no"`	

6.2.70 Router Element "<topic-resolution>"

Container for DRO topic resolution behavior options.

- **Cardinality:** 0 .. 1
- **Parent:** <endpoint>
- **Children:** <topic-use-query>, <pattern-use-query>, <remote-topic-interest>, <remote-pattern-interest>, <domain-route>, <initial-request>

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <initial-request>
          <rate-limit/>
        </initial-request>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.71 Router Element "<initial-request>"

Sets interval and duration for initial topic resolution requests.

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>

XML Attributes:

Attribute	Description	Valid Values	Default Value
periodic-interval	The interval at which the initial topic resolution requests are sent. Before changing the value of this option, please contact Informatica Support.	string	`"1000"`
duration	The minimum duration for which the initial topic resolution requests are sent. Before changing the value of this option, please contact Informatica Support.	string	`"10"`

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <initial-request duration="15" periodic-interval="800"/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
```

```
  </portals>
</tnw-gateway>
```

6.2.72 Router Element "<domain-route>"

Sets maximum and minimum limits for the interval between periodic domain route messages being sent for each remote domain that the portal is servicing.

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>
- **Children:** <rate-limit>

XML Attributes:

Attribute	Description	Valid Values	Default Value
min-interval	The minimum interval, in milliseconds, between domain route messages being sent for each domain.	string	`"100"`
max-interval	The maximum interval, in milliseconds, between domain route messages being sent for each domain.	string	`"1000"`

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <domain-route min-interval="100" max-interval="1000">
          <rate-limit bps="0" objects-per-second="50"/>
        </domain-route>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.73 Router Element "<rate-limit>"

Sets rate limits for topic resolution data sent over the network.

You can set rate limits individually for each of the topic resolution message types (see children elements).

- **Cardinality:** 0 .. 1
- **Parent:** <topic-use-query>, <pattern-use-query>, <remote-topic-interest>, <remote-pattern-interest>, <domain-route>

XML Attributes:

Attribute	Description	Valid Values	Default Value
bps	The limit in Bits per Second that data will be sent on the network. A value of 0 disables limiting by bits per second. Before changing the value of this option, please contact Informatica Support.	string	**"500000"** (For use queries and interest messages) **"0"** (For domain route messages)
objects-per-second	The limit in Objects per Second that data will be sent on the network. A value of 0 disables limiting by objects per second. Before changing the value of this option, please contact Informatica Support.	string	**"500"** (For use queries) **"0"** (For interest messages) **"50"** (For domain route messages)

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <topic-use-query max="6" periodic-interval="250000" timeout="4000"/>
          <rate-limit bps="550000" objects-per-second="0"/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.74 Router Element "<remote-pattern-interest>"

Sets parameters for when and how often this endpoint portal sends pattern interest messages

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>
- **Children:** <rate-limit>

XML Attributes:

Attribute	Description	Valid Values	Default Value
min-interval	The minimum interval, in milliseconds, between pattern interest messages being sent for each pattern the portal has interest in.	string	`"1000"`
max-interval	The maximum interval, in milliseconds, between pattern interest messages being sent for each pattern the portal has interest in.	string	`"60000"`

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <remote-pattern-interest min-interval="1000" max-interval="90000">
```

```
        <rate-limit/>
      </remote-pattern-interest>
    </topic-resolution>
  </endpoint>
 </portals>
</tnw-gateway>
```

6.2.75 Router Element "<remote-topic-interest>"

Sets parameters for when and how often this endpoint portal sends topic interest messages.

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>
- **Children:** <rate-limit>

XML Attributes:

Attribute	Description	Valid Values	Default Value
min-interval	The minimum interval, in milliseconds, between topic interest messages being sent for each topic the portal has interest in.	string	`"1000"`
max-interval	The maximum interval, in milliseconds, between topic interest messages being sent for each topic the portal has interest in.	string	`"60000"`

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <remote-topic-interest min-interval="1000" max-interval="90000">
          <rate-limit/>
        </remote-topic-interest>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.76 Router Element "<pattern-use-query>"

Sets parameters for when and how often this endpoint portal sends pattern use queries.

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>
- **Children:** <rate-limit>

XML Attributes:

Attribute	Description	Valid Values	Default Value
timeout	The maximum time, in milliseconds, to wait for a pattern use response. Before changing the value of this option, please contact Informatica Support.	string	"3000"
max	Maximum number of pattern use queries to send for a given pattern, each separated by the timeout value before giving up and removing the topic from the topic list. Before changing the value of this option, please contact Informatica Support.	string	"5"
periodic-interval	The interval, in milliseconds, between periodic pattern use queries being sent for each pattern the portal has interest in. Before changing the value of this option, please contact Informatica Support.	string	"300000"

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <pattern-use-query max="6" periodic-interval="250000" timeout="4000">
          <rate-limit/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.77 Router Element "<topic-use-query>"

Sets parameters for when and how often this endpoint portal sends topic use queries.

- **Cardinality:** 0 .. 1
- **Parent:** <topic-resolution>
- **Children:** <rate-limit>

XML Attributes:

Attribute	Description	Valid Values	Default Value
timeout	The maximum time, in milliseconds, to wait for a topic use response. Before changing the value of this option, please contact Informatica Support.	string	"3000"
max	Maximum number of topic use queries to send for a given topic, each separated by the timeout value before giving up and removing the topic from the topic list. Before changing the value of this option, please contact Informatica Support.	string	"5"
periodic-interval	The interval, in milliseconds, between periodic topic use queries being sent for each topic the portal has interest in. Before changing the value of this option, please contact Informatica Support.	string	"300000"

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <topic-resolution>
        <topic-use-query max="6" periodic-interval="250000" timeout="4000">
          <rate-limit/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.78 Router Element "<domain-id>"

Identifies the TRD for this endpoint portal. It must be unique within the DRO (which means that for any TRD, you can assign only one endpoint portal per DRO). Also, all endpoints interfacing a given TRD must have the same <domain-id> value.

There is no default, it must be supplied.

- **Parent:** <endpoint>

Example:

```
<tnw-gateway version="1.0">
  ...
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>1</cost>
      ...
    </endpoint>
  </portals>
</tnw-gateway>
```

6.2.79 Router Element "<daemon>"

Container for options common to the entire DRO process.

- **Cardinality:** 0 .. 1
- **Parent:** <tnw-gateway>
- **Children:** <name>, <log>, <uid>, <gid>, <pidfile>, <lbm-license-file>, <topicmap>, <patternmap>, <monitor>, <web-monitor>, <daemon-monitor>, <propagation-delay>, <xml-config>, <route-info>, <route-recalculation>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
  </daemon>
  ...
</tnw-gateway>
```

6.2.80 Router Element "<route-recalculation>"

Lets you set timing parameters for DRO rerouting route calculation behavior.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
backoff-interval	How long, in milliseconds, the DRO waits after the last detected change in topology before initiating a route recalculation.	string	"5000"
warning-interval	How long, in milliseconds, the DRO waits before warning that a route recalculation is being held up due to a non-converging topology.	string	"10000"

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <route-recalculation backoff-interval="5000" warning-interval="10000"/>
  </daemon>
  ...
</tnw-gateway>
```

6.2.81 Router Element "<route-info>"

Lets you set control parameters for DRO initial route setup (or reroute) behavior.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
propagation-interval	The time interval between route information messages that the DRO sends to other DRO.	string	"1000"
check-interval	How often the DRO checks to see if a route information message needs to be sent, a DRO has timed out, and/or the routes need to be recalculated.	string	"750"
timeout	How long a DRO waits after receiving no route information messages from another DRO before determining that that DRO is out of service or unreachable.	string	"4000"
max-hop-count	The maximum number of DROs a route information message can traverse before being discarded.	string	"100"

Example:

```
<tnw-gateway version="1.0">
  <daemon>
```

```
    ...
    <route-info propagation-interval="1000" check-interval="750" timeout="4000" max-hop-count="100"/>
  </daemon>
  ...
</tnw-gateway>
```

6.2.82 Router Element "<xml-config>"

Specifies the UM XML configuration file.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**`"default"`** - Trim whitespace. **`"preserve"`** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <xml-config>configfile.xml</xml-config>
  </daemon>
  ...
</tnw-gateway>
```

6.2.83 Router Element "<propagation-delay>"

DEPRECATED AND ELIMINATED. DO NOT USE. Specifies the difference between the shortest and longest propagation delays in the network.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
delta		string	

6.2.84 Router Element "<daemon-monitor>"

Configures the Daemon Statistics feature. See **Daemon Statistics** for general information on Daemon Statistics.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>
- **Children:** <lbm-config>, <publishing-interval>, <remote-snapshot-request>, <remote-config-changes-request>

XML Attributes:

Attribute	Description	Valid Values	Default Value
topic	Topic name to use for publishing Daemon Statistics.	string	`"tnwgd.monitor"`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <daemon-monitor topic="umrouter.1">
      <lbm-config>/path/umrouter_monitor.cfg</lbm-config>
      <publishing-interval>
        ...
      </publishing-interval>
      <remote-snapshot-request allow="1"/>
      <remote-config-changes-request allow="0"/>
    </daemon-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.85 Router Element "<remote-config-changes-request>"

Configures whether the DRO will respond to monitoring apps requests to change the rate at which Daemon Statistics messages are published. See **Daemon Statistics** for general information on Daemon Statistics.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon-monitor>

XML Attributes:

Attribute	Description	Valid Values	Default Value
allow	Enable or disable change requests.	`"0"` - Ignore change requests. `"1"` - Respond to change requests.	`"0"`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <daemon-monitor topic="umrouter.1">
      <lbm-config>/path/umrouter_monitor.cfg</lbm-config>
      <publishing-interval>
        ...
      </publishing-interval>
      <remote-snapshot-request allow="1"/>
```

```
      <remote-config-changes-request allow="0"/>
    </daemon-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.86 Router Element "<remote-snapshot-request>"

Configures whether the DRO will respond to monitoring apps requests to send on-demand snapshots of daemon statistics. See **Daemon Statistics** for general information on Daemon Statistics.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon-monitor>

XML Attributes:

Attribute	Description	Valid Values	Default Value
allow	Enable or disable snapshot requests.	`"0"` - Ignore snapshot requests. `"1"` - Respond to snapshot requests.	`"0"`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <daemon-monitor topic="umrouter.1">
      <lbm-config>/path/umrouter_monitor.cfg</lbm-config>
      <publishing-interval>
        ...
      </publishing-interval>
      <remote-snapshot-request allow="1"/>
      <remote-config-changes-request allow="0"/>
    </daemon-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.87 Router Element "<web-monitor>"

Identifies the address for the web monitor, in the form of interface:port. You can use "∗" to specify the local host.

Omit this element to disable the web monitor.

See **Webmon Security** for important security information.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <web-monitor>*:21001</web-monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.88 Router Element "<monitor>"

Container for UM Transport monitoring configuration elements.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>
- **Children:** <transport-module>, <format-module>

XML Attributes:

Attribute	Description	Valid Values	Default Value
interval	Monitoring interval, in seconds. 0 disables monitoring.	string	**"0"**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <monitor interval="30">
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="config=/cfgs/TD1.cfg;separator=|"/>
    </monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.89 Router Element "<format-module>"

Provides specifics about the monitoring format module.

- **Cardinality:** 0 .. 1
- **Parent:** <monitor>

XML Attributes:

Attribute	Description	Valid Values	Default Value
module	Selects the message formatting module.	**"csv"** - Comma-separated values (currently the only supported format).	**"csv"**
options	Option string to be passed to the formatting module. Available option is **"separator"** (defaults to comma).	string	(if omitted, no options are passed to the formatting module)

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <monitor interval="30">
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="separator=|"/>
    </monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.90 Router Element "<transport-module>"

Specifies characteristics about the monitoring transport module used.

- **Cardinality:** 0 .. 1
- **Parent:** <monitor>

XML Attributes:

Attribute	Description	Valid Values	Default Value
module	Selects the message transport module.	**"lbm"** - Publish messages via standard UM source. **"lbmsnmp"** - Publish messages via standard UM source with special settings intended for the UM SNMP agent. **"udp"** - Publish messages as simple UDP datagrams.	**"lbm"**
options	Option string to be passed to the transport module. Available options are **"config"** (configuration file pathname) and **"topic"** (the topic name to use for sending and receiving statistics; defaults to "/29west/statistics").	string	(if omitted, no options are passed to the transport module)

Example 1:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <monitor interval=30>
```

```
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="config=/cfgs/TD1.cfg;separator=|"/>
    </monitor>
  </daemon>
  ...
</tnw-gateway>
```

Example 2:

Monitoring configuration options can be supplied directly in the XML.

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <monitor interval=30>
      <transport-module module="lbm"
        options="config=/cfgs/TD1.cfg;context|request_tcp_interface=192.168.135.131"/>
      <format-module options="config=/cfgs/TD1.cfg;separator=|"/>
    </monitor>
  </daemon>
  ...
</tnw-gateway>
```

6.2.91 Router Element "<patternmap>"

Determines characteristics of the internal topic resolution maps for wildcard patterns.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
hash-function	Topic resolution hash function to use. Informatica recommends the default. See **resolver_string_↩ hash_function (context)** for more information.	**"classic"** - UM's original hash function. May be better for certain specialized topic names. **"djb2"** - The Dan Bernstein algorithm from comp.lang.c. May be better for topic names have a changing prefix with a constant suffix. **"sdbm"** - Sdbm database library (used in Berkeley DB). May be better for certain specialized topic names. **"murmur2"** - Good all-around hash function by Austin Appleby.	**"murmur2"**
size	Number of buckets in hash table. Should be a prime number.	string	**"131111"**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <patternmap hash-function="murmur2" size="131111">
  </daemon>
  ...
</tnw-gateway>
```

6.2.92 Router Element "<topicmap>"

Determines characteristics of the internal topic resolution maps for topic names.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
hash-function	Topic resolution hash function to use. Informatica recommends the default. See **resolver_string_↩ hash_function (context)** for more information.	**"classic"** - UM's original hash function. May be better for certain specialized topic names. **"djb2"** - The Dan Bernstein algorithm from comp.lang.c. May be better for topic names have a changing prefix with a constant suffix. **"sdbm"** - Sdbm database library (used in Berkeley DB). May be better for certain specialized topic names. **"murmur2"** - Good all-around hash function by Austin Appleby.	**"murmur2"**
size	Number of buckets in hash table. Should be a prime number.	string	**"131111"**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <topicmap hash-function="murmur2" size="131111">
  </daemon>
  ...
</tnw-gateway>
```

6.2.93 Router Element "<lbm-license-file>"

Specifies the UM license file's pathname.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
     . . .
    <lbm-license-file>lic0014.txt</lbm-license-file>
     . . .
  </daemon>
     . . .
     . . .
</tnw-gateway>
```

6.2.94 Router Element "<pidfile>"

Contains the pathname for daemon process ID (PID) file.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**`"default"`** - Trim whitespace. **`"preserve"`** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <pidfile>\var\run\dro01.pid</pidfile>
  </daemon>
  ...
</tnw-gateway>
```

6.2.95 Router Element "<gid>"

Specifies a Group ID (GID) for daemon process (if run as root).

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <gid>1234</gid>
  </daemon>
  ...
</tnw-gateway>
```

6.2.96 Router Element "<uid>"

Specifies a User ID (UID) for the daemon process (if run as root).

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <uid>5555</uid>
  </daemon>
  ...
</tnw-gateway>
```

6.2.97 Router Element "<log>"

Specifies the destination for DRO log messages. If you set the type for "file", use this element to contain the full pathname.

- **Cardinality:** 0 .. 1
- **Parent:** <daemon>

XML Attributes:

Attribute	Description	Valid Values	Default Value
type	Method of writing logs.	**"file"** - Write log to disk file. **"syslog"** - Write log to Unix "syslog". **"console"** - Write log to standard out.	**"console"**
frequency	Frequency by which to roll log file. Only applies for type="file".	**"disable"** - Do not roll log file. **"daily"** - Roll log file at midnight. **"hourly"** - Roll log file after approximately an hour, but is not exact and can drift significantly over a period of time. **"test"** - For Informatica internal use only. Do not use.	**"disable"**
size	Number of millions of bytes of file size to roll log file. E.g. a value of 1 rolls after 1000000 bytes. Maximum value is 4000. Value of 0 disables rolling by file size. Only applies for type="file".	string	**"0"**
xml:space	Specifies how whitespace (tabs, spaces, linefeeds) are handled in the element content. See **xml:space Attribute**.	**"default"** - Trim whitespace. **"preserve"** - Retain whitespace exactly as entered.	**default**

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    ...
    <log type="syslog"/>
  </daemon>
  ...
</tnw-gateway>
```

6.3 DRO Configuration DTD

Here is the XML configuration DTD with the comments removed. To see the DTD with comments included, enter `tnwgd --dump-dtd`.

```
<!ELEMENT tnw-gateway (daemon?, portals)>
<!ATTLIST tnw-gateway
          version (1.0) #REQUIRED
>
<!ELEMENT daemon (name?, log?, uid?, gid?, pidfile?, lbm-license-file?, topicmap?, patternmap?, monitor?,
      web-monitor?, daemon-monitor?, propagation-delay?, xml-config?, route-info?, route-recalculation?)>
<!ELEMENT log ( #PCDATA )>
<!ATTLIST log
          type (file | syslog | console) "console"
          frequency (disable | daily | hourly | test) "disable"
          size CDATA "0"
          xml:space (default | preserve) "default"
>
<!ELEMENT pidfile ( #PCDATA )>
<!ATTLIST pidfile xml:space (default | preserve) "default">
<!ELEMENT uid ( #PCDATA )>
<!ELEMENT gid ( #PCDATA )>
<!ELEMENT lbm-license-file ( #PCDATA )>
<!ATTLIST lbm-license-file xml:space (default | preserve) "default">
<!ELEMENT topicmap EMPTY>
<!ATTLIST topicmap
          hash-function ( classic | djb2 | sdbm | murmur2 ) "murmur2"
          size CDATA "131111"
>
<!ELEMENT patternmap EMPTY>
<!ATTLIST patternmap
          hash-function ( classic | djb2 | sdbm | murmur2 ) "murmur2"
          size CDATA "131111"
>
<!ELEMENT portals (endpoint | peer)+>
<!ELEMENT endpoint (name, domain-id, cost?, source-deletion-delay?, max-queue?, smart-batch?, lbm-config?,
      lbm-attributes?, acl?, topic-resolution?, late-join?, topic-purge?, topic-interest-generate?, topic-domain-
      activity?, pattern-purge?, pattern-interest-generate?, pattern-domain-activity?, remote-topic?, remote-
      pattern?, source-context-name?, receiver-context-name?, sqn-window?, context-query?, publishing-interval? )>
<!ELEMENT peer (name, cost?, sourcemap?, (tcp | single-tcp), source-deletion-delay?, max-queue?, smart-
      batch?, max-datagram?, batching?, lbm-config?, lbm-attributes?, acl?, topic-purge?, topic-interest-generate?,
      topic-domain-activity?, pattern-purge?, pattern-interest-generate?, pattern-domain-activity?, topic-use-check?
      , pattern-use-check?, source-context-name?, receiver-context-name?, sqn-window?, context-query?, gateway-
      keepalive?, publishing-interval? )>
<!ELEMENT name ( #PCDATA )>
<!ATTLIST name xml:space (default | preserve) "default">
<!ELEMENT domain-id ( #PCDATA )>
<!ELEMENT cost ( #PCDATA )>
<!ELEMENT source-deletion-delay ( #PCDATA )>
<!ELEMENT sourcemap EMPTY>
<!ATTLIST sourcemap
          size CDATA "131072"
>
<!ELEMENT tcp (interface?, listen-port, receive-buffer?, send-buffer?, keepalive?, nodelay?, compression?,
      tls?, companion )>
<!ELEMENT interface ( #PCDATA )>
<!ELEMENT listen-port ( #PCDATA )>
<!ELEMENT receive-buffer ( #PCDATA )>
<!ELEMENT send-buffer ( #PCDATA )>
<!ELEMENT keepalive EMPTY >
<!ELEMENT nodelay EMPTY >
<!ELEMENT companion (address, port)>
<!ELEMENT compression ( #PCDATA )>
<!ELEMENT tls (certificate, certificate-key, certificate-key-password?, trusted-certificates?, cipher-
      suites? )>
<!ELEMENT certificate ( #PCDATA )>
<!ELEMENT certificate-key ( #PCDATA )>
<!ELEMENT certificate-key-password ( #PCDATA )>
<!ELEMENT trusted-certificates ( #PCDATA )>
```

```
<!ELEMENT cipher-suites ( #PCDATA )>
<!ATTLIST companion
          reconnect-interval CDATA "5000"
>
<!ELEMENT address ( #PCDATA )>
<!ELEMENT port ( #PCDATA )>
<!ELEMENT single-tcp (interface?, receive-buffer?, send-buffer?, keepalive?, nodelay?, compression?, tls?,
      (initiator | acceptor ) )>
<!ELEMENT initiator (address, port)>
<!ATTLIST initiator
          reconnect-interval CDATA "5000"
>
<!ELEMENT acceptor (listen-port)>
<!ELEMENT max-queue ( #PCDATA )>
<!ELEMENT smart-batch ( #PCDATA )>
<!ELEMENT max-datagram ( #PCDATA )>
<!ELEMENT batching (min-length?, batch-interval?)>
<!ELEMENT min-length ( #PCDATA )>
<!ELEMENT batch-interval ( #PCDATA )>
<!ELEMENT lbm-config ( #PCDATA )>
<!ATTLIST lbm-config xml:space (default | preserve) "default">
<!ELEMENT lbm-attributes (option+)>
<!ELEMENT option EMPTY>
<!ATTLIST option
          scope (receiver | context | source | wildcard_receiver | event_queue) #REQUIRED
          name CDATA #REQUIRED
          value CDATA #REQUIRED
>
<!ELEMENT acl (inbound?, outbound?)>
<!ELEMENT inbound (ace+)>
<!ELEMENT outbound (ace+)>
<!ELEMENT ace (topic | pcre-pattern | regex-pattern | transport | source-ip | multicast-group | udp-source-
      port | udp-destination-port | tcp-source-port | xport-id)+ >
<!ATTLIST ace match (accept | reject) #REQUIRED >
<!ELEMENT topic ( #PCDATA )>
<!ATTLIST topic
          xml:space (default | preserve) "default"
>
<!ELEMENT pcre-pattern ( #PCDATA )>
<!ATTLIST pcre-pattern
          xml:space (default | preserve) "default"
>
<!ELEMENT regex-pattern ( #PCDATA )>
<!ATTLIST regex-pattern
          xml:space (default | preserve) "default"
>
<!ELEMENT transport EMPTY>
<!ATTLIST transport
          value (tcp | lbt-rm | lbtrm | lbt-ru | lbtru | lbt-ipc | lbtipc) #REQUIRED
          comparison (eq | equal | ne | notequal) #REQUIRED
>
<!ELEMENT source-ip EMPTY>
<!ATTLIST source-ip
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
<!ELEMENT multicast-group EMPTY>
<!ATTLIST multicast-group
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
<!ELEMENT udp-source-port EMPTY>
<!ATTLIST udp-source-port
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
<!ELEMENT udp-destination-port EMPTY>
<!ATTLIST udp-destination-port
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
<!ELEMENT tcp-source-port EMPTY>
<!ATTLIST tcp-source-port
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
<!ELEMENT xport-id EMPTY>
<!ATTLIST xport-id
          value CDATA #REQUIRED
          comparison (eq | equal | ne | notequal | lt | lessthan | le | lessthanequal | gt | greaterthan |
     ge | greaterthanequal) #REQUIRED
>
```

```
<!ELEMENT topic-resolution (topic-use-query?, pattern-use-query?, remote-topic-interest?, remote-pattern-
      interest?, domain-route?, initial-request? )>
<!ELEMENT topic-use-query (rate-limit? ) >
<!ATTLIST topic-use-query
          timeout CDATA "3000"
          max CDATA "5"
          periodic-interval CDATA "300000"
>
<!ELEMENT pattern-use-query (rate-limit? ) >
<!ATTLIST pattern-use-query
          timeout CDATA "3000"
          max CDATA "5"
          periodic-interval CDATA "300000"
>
<!ELEMENT remote-topic-interest (rate-limit? ) >
<!ATTLIST remote-topic-interest
          min-interval CDATA "1000"
          max-interval CDATA "60000"
>
<!ELEMENT remote-pattern-interest (rate-limit? ) >
<!ATTLIST remote-pattern-interest
          min-interval CDATA "1000"
          max-interval CDATA "60000"
>
<!ELEMENT domain-route (rate-limit? ) >
<!ATTLIST domain-route
          min-interval CDATA "100"
          max-interval CDATA "1000"
>
<!ELEMENT rate-limit EMPTY>
<!ATTLIST rate-limit
          bps CDATA #IMPLIED
          objects-per-second CDATA #IMPLIED
>
<!ELEMENT initial-request EMPTY>
<!ATTLIST initial-request
          periodic-interval CDATA "1000"
          duration CDATA "10"
>
<!ELEMENT late-join EMPTY>
<!ATTLIST late-join
          provide ( source | always | never ) "source"
          forward ( yes | no ) "yes"
>
<!ELEMENT topic-purge EMPTY>
<!ATTLIST topic-purge periodic-interval CDATA #IMPLIED>
<!ELEMENT topic-interest-generate EMPTY>
<!ATTLIST topic-interest-generate
          periodic-interval CDATA #IMPLIED
          max-topics CDATA #IMPLIED
          interval CDATA #IMPLIED
>
<!ELEMENT topic-domain-activity EMPTY>
<!ATTLIST topic-domain-activity timeout CDATA #IMPLIED>
<!ELEMENT pattern-purge EMPTY>
<!ATTLIST pattern-purge periodic-interval CDATA #IMPLIED>
<!ELEMENT pattern-interest-generate EMPTY>
<!ATTLIST pattern-interest-generate
          periodic-interval CDATA #IMPLIED
          max-patterns CDATA #IMPLIED
          interval CDATA #IMPLIED
>
<!ELEMENT pattern-domain-activity EMPTY>
<!ATTLIST pattern-domain-activity timeout CDATA #IMPLIED>
<!ELEMENT remote-topic EMPTY>
<!ATTLIST remote-topic
          check-interval CDATA "90000"
          max-topics CDATA "100"
          timeout CDATA "300000"
>
<!ELEMENT remote-pattern EMPTY>
<!ATTLIST remote-pattern
          check-interval CDATA "90000"
          max-patterns CDATA "100"
          timeout CDATA "300000"
>
<!ELEMENT topic-use-check EMPTY>
<!ATTLIST topic-use-check periodic-interval CDATA #IMPLIED>
<!ELEMENT pattern-use-check EMPTY>
<!ATTLIST pattern-use-check periodic-interval CDATA #IMPLIED>
<!ELEMENT monitor (transport-module?, format-module?)>
<!ATTLIST monitor
          interval CDATA "0"
>
<!ELEMENT transport-module EMPTY>
<!ATTLIST transport-module
          module (lbm | lbmsnmp | udp) "lbm"
```

```
          options CDATA #IMPLIED
>
<!ELEMENT format-module EMPTY>
<!ATTLIST format-module
          module (csv) "csv"
          options CDATA #IMPLIED
>
<!ELEMENT web-monitor ( #PCDATA )>
<!ATTLIST web-monitor xml:space (default | preserve) "default">
<!ELEMENT propagation-delay EMPTY>
<!ATTLIST propagation-delay delta CDATA #IMPLIED>
<!ELEMENT xml-config ( #PCDATA )>
<!ATTLIST xml-config xml:space (default | preserve) "default">
<!ELEMENT source-context-name ( #PCDATA )>
<!ATTLIST source-context-name xml:space (default | preserve) "default">
<!ELEMENT receiver-context-name ( #PCDATA )>
<!ATTLIST receiver-context-name xml:space (default | preserve) "default">
<!ELEMENT sqn-window EMPTY>
<!ATTLIST sqn-window
          size CDATA "16384"
          increment CDATA "2048"
>
<!ELEMENT context-query EMPTY>
<!ATTLIST context-query
          periodic-interval CDATA #IMPLIED
          max-contexts CDATA #IMPLIED
          interval CDATA #IMPLIED
          timeout CDATA #IMPLIED
>
<!ELEMENT gateway-keepalive EMPTY>
<!ATTLIST gateway-keepalive
          idle ( yes | no ) "yes"
          interval CDATA "5000"
          timeout CDATA "15000"
>
<!ELEMENT route-info EMPTY>
<!ATTLIST route-info
          propagation-interval CDATA "1000"
          check-interval CDATA "750"
          timeout CDATA "4000"
          max-hop-count CDATA "100"
>
<!ELEMENT route-recalculation EMPTY>
<!ATTLIST route-recalculation
          backoff-interval CDATA "5000"
          warning-interval CDATA "10000"
>
<!ELEMENT daemon-monitor (lbm-config?, publishing-interval?, remote-snapshot-request?, remote-config-
      changes-request?)>
<!ATTLIST daemon-monitor topic CDATA "tnwgd.monitor">
<!ELEMENT publishing-interval (group+)>
<!ELEMENT group EMPTY>
<!ATTLIST group name (default | gateway-config | route-manager-topology | malloc-info | portal-config |
      portal-stats ) #REQUIRED>
<!ATTLIST group ivl CDATA #REQUIRED>
<!ELEMENT remote-snapshot-request EMPTY>
<!ATTLIST remote-snapshot-request allow (0 | 1) "0">
<!ELEMENT remote-config-changes-request EMPTY>
<!ATTLIST remote-config-changes-request allow (0 | 1) "0">
```

Chapter 7

DRO Daemon Statistics

This section contains details on the DRO's Daemon Statistics feature. **You should already be familiar with the general information contained in Daemon Statistics.**

7.1 DRO Daemon Statistics Structures

The different message types are:

- **TNWG_DSTATTYPE_MALLINFO**
- **TNWG_DSTATTYPE_GATEWAYCFG**
- **TNWG_DSTATTYPE_PORTCFG**
- **TNWG_DSTATTYPE_RM_LOCAL**
- **TNWG_DSTATTYPE_RM_PORTAL**
- **TNWG_DSTATTYPE_RM_OTHERGW**
- **TNWG_DSTATTYPE_RM_OTHERGW_NBR**
- **TNWG_DSTATTYPE_PORTSTAT**

Each one has a specific structure associated with it, as detailed in the file **tnwgdmonmsgs.h**.

Note that message types ending with "CFG" are in the config category. All others are in the stats category. See **Daemon Statistics Structures** for information on how the two categories are handled differently.

7.1.1 DRO Daemon Statistics Byte Swapping

A monitoring application receiving these messages must detect if there is an endian mismatch (see **Daemon Statistics Binary Data**). The header structure **tnwg_dstat_msg_hdr_t** contains a 16-bit field named `magic` which is set equal to **LBM_TNWG_DAEMON_MAGIC**. The receiving application should compare it to **LBM_TNWG_DAE↩MON_MAGIC** and **LBM_TNWG_DAEMON_ANTIMAGIC**. Anything else would represent a serious problem.

If the receiving app sees:

```
magic == LBM_TNWG_DAEMON_MAGIC
```

then it can simply access the binary fields directly. However, if it sees:

```
magic == LBM_TNWG_DAEMON_ANTIMAGIC
```

then *most* (but not all) binary fields need to be byte-swapped. See tnwgdmon.c for an example, paying special attention to the macros COND_SWAPxx (which *conditionally* swaps based on the magic test) and the functions byte_swapXX() (which performs the byte swapping).

7.1.2 DRO Daemon Statistics String Buffers

DRO Daemon Statistics data structures sometimes contain string buffers. Strings in these data structures are always null-terminated. These messages are generally sent as fixed-length equal to the sizes of the structures, and therefore include all of the declared bytes of the string fields, even if the contained string uses fewer bytes than declared. For example, the structure **tnwg_dstat_record_hdr_t** contains the field **tnwg_dstat_record_hdr_t_stct::portal_name** which is a char array of size TNWG_DSTAT_MAX_PORTAL_NAME_LEN. If portal_name is set to "p1", then only 3 bytes of the buffer are used (including the null string terminator). However, all TNWG_DSTAT_MAX_PORTAL_NAME_LEN bytes will be sent in the **TNWG_DSTATTYPE_RM_PORTAL** message type.

Contrast this with **Store Daemon Statistics String Buffers**.

There are two exceptions to this rule: **TNWG_DSTATTYPE_PORTCFG** and **TNWG_DSTATTYPE_GATEWAYCFG**.

The **TNWG_DSTATTYPE_PORTCFG** message is of type **tnwg_pcfg_stat_grp_msg_t** and has the field **tnwg_pcfg_stat_grp_msg_t_stct::data**. This field is a variable-length string buffer which contains one or more null-terminated strings. The total length of the **TNWG_DSTATTYPE_PORTCFG** message is the sum of the length of its sub-structures plus the number of bytes of string data (characters plus string-terminating nulls). The number of strings in **tnwg_pcfg_stat_grp_msg_t_stct::data** is given by **tnwg_pcfg_stat_grp_msg_t_stct::rechdr**->num_options. The monitoring application must step through the string buffer that many times to find each string. For an example of how to do this, see tnwgdmon.c in the code following, "'case TNWG_DSTATTYPE_PORTCFG:'".

The **TNWG_DSTATTYPE_GATEWAYCFG** message is of type **tnwg_dstat_gatewaycfg_msg_t** and has the field **tnwg_dstat_gatewaycfg_msg_t_stct::data**. This field is a variable-length string buffer which contains exactly one null-terminated string. This string contains the entirety of the DRO's configuration file. The individual lines contain the normal line-ending character(s). The total length of the **TNWG_DSTATTYPE_GATEWAYCFG** message is the length of its sub-structure plus the number of bytes of string data (characters plus string-terminating nulls).

7.2 DRO Daemon Statistics Configuration

There are three places in the DRO configuration file that Daemon Statistics are configured:

- The <daemon-monitor> element inside the <daemon> definition. Configures all aspects of the DRO Daemon Statistics feature, including publishing intervals.
- The <publishing-interval> element inside the <peer> definition. Configures only the publishing intervals on a peer portal basis.
- The <publishing-interval> element inside the <endpoint> definition. Configures only the publishing intervals on an endpoint portal basis.

Here is an example of configuring daemon statistics.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- G1 xml file- 2 endpoint portals -->
<tnw-gateway version="1.0">
  <daemon>
    ...
    <publishing-interval>
      <group name="default" ivl="3"/>
      <group name="gateway-config" ivl="120"/>
      <group name="portal-config" ivl="120"/>
    </publishing-interval>
    <remote-snapshot-request allow="1"/>
    <remote-config-changes-request allow="1"/>
  </daemon>
  <portals>
    <endpoint>
      <name>G1-TRD1</name>
      ...
      <publishing-interval>
        <group name="default" ivl="6"/>
        <group name="gateway-config" ivl="120"/>
        <group name="portal-config" ivl="120"/>
      </publishing-interval>
    </endpoint>
    ...
  </portals>
</tnw-gateway>
```

In this example, all stats-type messages are (conditionally) published on a 3-second interval, except those of portal G1-TRD1, which are published (conditionally) on a 6-second interval. All config-type messages are published (unconditionally) on a 120-second interval.

7.3 DRO Daemon Control Requests

The DRO Daemon supports a monitoring application to send a specific set of requests to control the operation of Daemon Statistics. The <remote-snapshot-request> and <remote-config-changes-request> configuration elements control whether the DRO enables the **Daemon Controller** operation (defaults to disabled).

Warning

> If misused, the Daemon Control Requests feature allows a user to interfere with the messaging infrastructure in potentially disruptive ways. By default, this feature is disabled. However, especially if you have enabled **UMP Element "<remote-config-changes-request>"**, Informatica recommends **Securing Daemon Control Requests**.

If enabled, the monitoring application can send a command message to the DRO in the form of a topicless unicast immediate "request" message (see **lbm_unicast_immediate_request()** with NULL for topic). The format of the message is a simple ascii string, with or without null termination. Due to the simple format of the message, no data structure is defined for it.

When the DRO receives and validates the command, it sends a UM response message back to the requesting application containing a status message (which is *not* null-terminated). If the status was OK, the DRO also performs the requested action.

7.3.1 DRO Daemon Control Request Addressing

Since Daemon Control Requests are sent as UIM messages, you must use a target string to address the request to the desired DRO Process. The general form of a UIM target address is described in **UIM Addressing**, but is illustrated by this example:

```
TCP:10.29.3.46:12009
```

where 10.29.3.46:12009 is the IP and Port of the Daemon Control context UIM port. These are typically configured using the **request_tcp_interface (context)** and **request_tcp_port (context)** options in the UM configuration file specified by the Router Element "<lbm-config>" contained within the Router Element "<daemon-monitor>".

7.3.2 DRO Control Request Types

The example program `tnwgdcmd.c` demonstrates the correct way to send the messages and receive the responses.

REQUEST TYPES ENABLED BY <remote-snapshot-request>:

version

The DRO returns in its command response the value of **LBM_UMESTORE_DMON_VERSION**. No daemon statistics messages are published.

snap mallinfo

The DRO immediately publishes the memory allocation usage message of type **TNWG_DSTATTYPE_MAL↩LINFO**.

snap pstat

The DRO immediately publishes the portal statistics message(s) of type **TNWG_DSTATTYPE_PORTSTAT**.

snap ri

The DRO immediately publishes the route information message(s) of types **TNWG_DSTATTYPE_RM_LO↩CAL**, **TNWG_DSTATTYPE_RM_PORTAL TNWG_DSTATTYPE_RM_OTHERGW**, and **TNWG_DSTATTY↩PE_RM_OTHERGW_NBR**.

snap gcfg

The DRO immediately publishes the gateway configuration message **TNWG_DSTATTYPE_GATEWAYCFG**.

snap pcfg

The DRO immediately publishes the portal configuration message(s) **TNWG_DSTATTYPE_PORTCFG**.

REQUEST TYPES ENABLED BY <remote-config-changes-request>:

mallinfo N

Set the publishing interval for memory allocation usage.
For example: `mallinfo 5`

ri N

Set the publishing interval for the route information messages.
For example: `ri 5`

gcfg N

Set the publishing interval for the gateway configuration message.
For example: `gcfg 5`

pstat N

Set the publishing interval for the portal statistics messages. This command can be preceded by a portal name in double quote marks to only set the publishing interval for that portal.
For example: `"G1-TRD1" pstat 5`

pcfg N

Set the publishing interval for the portal configuration messages. This command can be preceded by a portal name in double quote marks to only set the publishing interval for that portal.
For example: `"G1-TRD1" pcfg 5`

Chapter 8

DRO Monitoring

8.1 DRO Web Monitor

The built-in web monitor (configured in the `tnwgd` XML configuration file; see XML Configuration Reference) provides valuable statistics about the DRO and its portals, for which, the Web Monitor separates into receive statistics and send statistics. The Web Monitor provides a page for each endpoint and peer portal.

Warning

> The DRO's web monitor is not designed to be a highly-secure feature. Anybody with access to the network can access the web monitor pages.

Users are expected to prevent unauthorized access to the web monitor through normal firewalling methods. Users who are unable to limit access to a level consistent with their overall security needs should disable the DRO web monitor (using <web-monitor>). See **Webmon Security** for more information.

Note

> the UM daemon designs are evolving away from simple web-based monitoring and towards a publish/subscribe model of distributing monitoring events and statistics.

8.1.1 Main Page

This page displays general information about the DRO, and also provides the following links to more detailed statistical and configuration information.

UM Router Configuration

> Displays the DRO XML configuration file used by this DRO.

Portals

> Displays portal statistics and information, one portal per page. The Portals page allows you to link to any of the Peer or Endpoint portals configured for the DRO.

Topology Info

This links to a page that displays DRO network connectivity information from the perspective of this DRO.

Path Info

This lets you query and display a hop path that messages will take between any two TRDs.

On some platforms, the Main page may include a link (GNU malloc info) to a memory allocation display page that displays the following:

arena

Non-mmapped space allocated (bytes)

ordblks

Number of free chunks

hblks

Number of mmapped regions

hblkhd

Space allocated in mmapped regions (bytes)

uordblks

Total allocated space (bytes)

fordblks

Total free space (bytes)

8.1.2 Endpoint Portal Page

The Endpoint Portal Page displays Receive and Send statistics for the selected endpoint portal. Receive statistics pertain to messages entering the portal from its connected TRD. Send statistics pertain to messages sent out to the TRD.

Click on any of the links at the top of the page to review configuration option values for the portal's UM topic resolution domain. The two columns provide different units of measure for a given statistic type, where the first column is typically in fragments or messages (depending on the statistic type), and the second column is in bytes.

Endpoint Portal *name*

Domain ID

The ID for the Topic Resolution Domain (TRD) to which this portal is connected.

Portal Cost

The cost value assigned to this portal.

Local Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers in the immediately adjacent TRD.

Topics

Of the local interest total, the number of topics.

PCRE patterns

Of the local interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the local interest total, the number of wildcard patterns, using REGEX pattern matching.

Remote Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers beyond and downstream from the immediately adjacent TRD.

Topics

Of the remote interest total, the number of topics.

PCRE patterns

Of the remote interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the remote interest total, the number of wildcard patterns, using REGEX pattern matching.

Proxy Receivers

The number of proxy receivers active in this portal.

Receiver Topics

The number of topics in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Receiver PCRE patterns

The number of wildcard patterns, using PCRE pattern matching, in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Receiver REGEX patterns

The number of wildcard patterns, using REGEX pattern matching, in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Proxy Sources

The number of proxy sources active in this portal.

Endpoint Receive Statistics

Transport topic fragments/bytes received

The total transport-based topic-related traffic of messages containing user data received by this portal from a TRD. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic request fragments/bytes received

Topic messages received that are request messages, i.e., messages send via lbm_send_request∗() rather than lbm_src_send∗().

Transport topic control msgs/bytes received

The total transport-based topic-related traffic received by this portal from a TRD. These are supervisory messages, which include TSNIs, SRIs., etc. The first column counts the number of messages.

Immediate topic fragments/bytes received

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), that have a topic, received at this portal.

Immediate topic request fragments/bytes received

Of the MIM topic messages received, this is the amount of those that are requests.

Immediate topicless fragments/bytes received

The total number of MIM messages or message fragments, and bytes (second column), with null topics, received by his portal.

Immediate topicless request fragments/bytes received

Of the MIM topicless messages received, this is the amount of those that are requests.

Unicast data messages/bytes received

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column) containing user data, received by this portal.

Duplicate unicast data messages/bytes dropped

UIM data messages discarded because they were duplicates of messages already received.

Unicast data messages/bytes received with no stream info

UIM data messages discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast data messages/bytes received with no route to destination

UIM data messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Unicast control messages/bytes received

The total number of Unicast Immediate Messaging (UIM) supervisory (non-data) messages (and bytes, second column) received by this portal.

Duplicate unicast control messages/bytes dropped

Supervisory UIMs dropped because they were duplicates of messages already received.

Unicast control messages/bytes received with no stream info

Supervisory UIMs dropped because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast control messages/bytes received with no route to destination

Supervisory UIM messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Endpoint Send Statistics

Transport topic fragments/bytes forwarded

The total transport-based topic-related traffic forwarded to this portal from other portals in this DRO. This could include user messages, TSNIs, SRIs, etc. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragments/bytes sent

Of the transport topic traffic forwarded, this is the amount of traffic sent out to the TRD.

Transport topic request fragments/bytes sent

Of the messages sent, this is the amount of those that are requests.

Duplicate transport topic fragments/bytes dropped

Of the messages forwarded to this portal, this is the total of those that were discarded because they were duplicates of messages already received.

Transport topic fragments/bytes dropped due to blocking

Of the messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Transport topic fragments/bytes dropped due to error

Of the messages forwarded to this portal, this is the total of those that were discarded due to an application or network connection failure.

Transport topic fragments/bytes dropped due to fragment size error

Of the messages forwarded to this portal, this is the total of those that were discarded possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Immediate topic fragments/bytes forwarded

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), forwarded to this portal from other portals in this DRO.

Immediate topic fragments/bytes sent

Of the MIM topic messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Immediate topic request fragments sent

Of the MIM topic messages sent, this is the amount of those that are requests.

Immediate topic fragments/bytes dropped due to blocking

Of the MIM topic messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Immediate topic fragments/bytes dropped due to error

Of the MIM topic messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Immediate topic fragments/bytes dropped due to fragment size error

Of the MIM topic messages forwarded to this portal, those that were dropped possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Immediate topicless fragments/bytes forwarded

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), with null topics, forwarded to this portal from other portals in this DRO.

Immediate topicless fragments/bytes sent

Of the MIM topicless messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Immediate topicless request fragments sent

Of the MIM topicless messages sent, this is the amount of those that are requests.

Immediate topicless fragments/bytes dropped due to blocking

Of the MIM topicless messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Immediate topicless fragments/bytes dropped due to error

Of the MIM topicless messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Immediate topicless fragments/bytes dropped due to fragment size error

Of the MIM topicless messages forwarded to this portal, those that were dropped possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Unicast messages/bytes forwarded

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column), both control and containing user data, forwarded to this portal.

Unicast messages/bytes sent

Of the UIM data messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Unicast messages/bytes dropped due to error

Of the UIM data messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Current/maximum data bytes enqueued (limit: **n**)

For bytes in this portal's send buffer (due to a blocking send), the first column is a snapshot of the current amount, and the second column is a high-water mark. The displayed limit (n) is the configuration value for option <max-queue>.

8.1.3 Peer Portal Page

This page allows you to see Receive and Send statistics for the selected peer portal. Click on any of the links at the top of the page to review configuration option values for the portal's UM topic resolution domain.

The peer portal page displays the following statistics:

Peer Portal *name*

Portal Cost

The cost value assigned to this portal.

Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers beyond and downstream from the immediately adjacent DRO.

Topics

Of the interest total, the number of topics.

PCRE patterns

Of the interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the interest total, the number of wildcard patterns, using REGEX pattern matching.

Proxy Receivers

The number of proxy receivers active in this portal.

Receiver topics

All topics in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Receiver PCRE patterns

All wildcard patterns, using PCRE pattern matching, in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Receiver REGEX patterns

All wildcard patterns, using REGEX pattern matching, in which the other portals in the DRO have detected current interest and summarily propagated to this portal.

Proxy Sources

The number of proxy sources active in this portal.

Peer Receive Statistics

Data messages/bytes received

The total of messages containing data received at this portal. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragment data messages/bytes received

The total of user-data messages received on any topic resolved through this portal. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragment data messages/bytes received with unknown source

Topic messages received whose source this DRO has not seen before.

Transport topic request fragment data messages/bytes received

These are topic messages received that are request messages, i.e., messages send via lbm_send_request∗() rather than lbm_src_send∗().

Transport topic request fragment data messages/bytes received with unknown source

Of the request messages received, the topic messages received whose source this DRO has not seen before.

Immediate topic fragments/bytes received

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), that have a topic, received by all proxy receivers at this portal.

Immediate topic request fragments/bytes received Of the MIM topic messages received, this is the total of those that are requests.

Immediate topicless fragments/bytes received

The total number of MIM messages or message fragments, and bytes (second column), with null topics, received by all proxy receivers at this portal.

Immediate topicless request fragments/bytes received

Of the MIM topicless messages received, this is the total of those that are requests.

Unicast data messages/bytes received

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column) containing user data, received by this portal.

Unicast data messages/bytes received with no stream information

UIM data messages discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast data messages/bytes received with no route to destination

UIM data messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Control messages/bytes received

The total of supervisory messages (containing no data) received at this portal.

Transport topic control messages/bytes received

Of the control messages received, those that are transport/topic based (such as TSNIs, SRIs., etc.).

Transport topic control messages/bytes received with unknown source

Of the transport/topic control messages received whose source this DRO has not seen before.

Unicast control messages/bytes received

The total number of Unicast Immediate Messaging (UIM) supervisory (non-data) messages (and bytes, second column) received by this portal.

Retransmission requests/bytes received

Supervisory UIMs that are requests for retransmission of lost (or Late Join) messages.

Control messages/bytes received with no stream info

Supervisory UIMs discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Control messages/bytes received with no route to destination

Supervisory UIM messages that are on a wrong path, possibly due to a route recalculation.

Gateway control messages/bytes received

The total of DRO-only, peer-to-peer supervisory messages received at this portal.

Unhandled control messages/bytes received

Supervisory UIMs discarded because, though they are well-formed, they have no valid action request. This tally should stay at 0; otherwise, contact Informatica Support.

Peer Send Statistics

Transport topic fragments/bytes forwarded

The total transport-based topic-related traffic forwarded to this portal from other portals in this DRO. This could include user messages, TSNIs, SRIs., etc. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragments/bytes sent

Of transport topic messages forwarded to this portal, the amount of traffic sent to the adjacent DRO.

Transport topic request fragments/bytes sent

Of transport topic messages sent, those that were request messages.

Transport topic fragments/bytes dropped (duplicate)

Of transport topic messages forwarded to this portal, messages discarded because they were duplicates of messages already received.

Transport topic fragments/bytes dropped (blocking)

Of transport topic messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The DRO's XML configuration file may need to be adjusted.

Transport topic fragments/bytes dropped (not **operational**)

Of transport topic messages forwarded to this portal, messages discarded because the peer link is down.

Transport topic fragments/bytes dropped (queue **failure**)

Of transport topic messages forwarded to this portal, messages discarded due to a memory allocation failure.

Unicast messages/bytes forwarded

The total number of supervisory (no data payloads) Unicast Immediate Messaging (UIM) messages (and bytes, second column) forwarded to this portal from other portals in this DRO. These messages can be either control (supervisory) messages or contain user data.

Unicast messages/bytes sent

Of the UIMs forwarded to this portal, the amount of traffic sent to the adjacent DRO.

Unicast messages/bytes dropped (blocking)

Of the UIMs forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The DRO's XML configuration file may need to be adjusted.

Unicast messages/bytes dropped (not **operational**)

Of the UIMs forwarded to this portal, messages discarded because the peer link is down.

Unicast messages/bytes dropped (queue **failure**)

Of the UIMs forwarded to this portal, messages discarded due to a memory allocation failure.

Gateway control messages/bytes sent

The total number of DRO supervisory messages (and bytes, second column), generated at this portal.

Gateway control messages/bytes sent Of the DRO supervisory messages generated, the number sent to the adjacent DRO.

Gateway control messages/bytes dropped (blocking) The amount of DRO supervisory messages that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The DRO's XML configuration file may need to be adjusted.

Gateway control messages/bytes dropped (not **operational**)

The amount of DRO supervisory messages that were discarded because the peer link was down.

Gateway control messages/bytes dropped (queue **failure**)

The amount of DRO supervisory messages that were discarded due to a memory allocation failure.

Batches

The number of times messages were batched.

Minimum messages/bytes per batch

The lowest recorded number of messages in a batch, and the number of bytes in that batch.

Average messages/bytes per batch

The average number of messages in a batch, and the number of bytes in that average batch.

Maximum messages/bytes per batch

The highest recorded number of messages in a batch, and the number of bytes in that batch.

Current/maximum data bytes enqueued

For bytes in this portal's send buffer (due to a blocking send), the first column is a snapshot of the current amount, and the second column is a high-water mark. The displayed limit is the configuration value for option `<max-queue>`.

Keepalive/RTT samples

The number of keepalive messages that have been set to the other DRO's portal and responded to.

Minimum RTT (microseconds)

Of the keepalives sent and responded to, the lowest recorded round-trip time.

Mean RTT (microseconds)

Of the keepalives sent and responded to, the mean recorded round-trip time.

Maximum RTT (microseconds)

Of the keepalives sent and responded to, the highest recorded round-trip time.

Last keepalive responded to

The send timestamp (date and time) of the last sent keepalive message that was responded to.

8.1.4 Topology Info Page

This page allows you to see DRO network connectivity information from the perspective of this DRO. The *Other DROs* section (below) provides information in the same format as is used for the local DRO.

Local UM Router Name

The DRO name as assigned via configuration.

Local UM Router ID

A unique value that the DRO assigns to itself automatically.

Self Version

A configuration version for this DRO, as seen collectively by the DRO network.

Topology Signature

An identifier for the "map" of this DRO network's routes. This value should be the same for all DROs.

Last recalc duration

The amount of time in seconds that it took this DRO to perform its most recent route recalculation.

Graph Version

The number of times this DRO has updated its view of the topology.

UM Router Count

The number of DROs in this DRO network.

Topic Resolution Domain Count

The number of TRDs in this DRO network.

Portal (endpoint or peer)

This display is repeated for each portal of this DRO.

Portal Name

The portal's name as assigned via configuration.

Adjacent Domain/UM Router ID

For an endpoint portal, this is the configured <domain-id> for the connected TRD. For a peer portal, this is an automatically assigned unique identifier for the connected DRO.

Cost

This portal's configured cost.

Last interest recalc duration

The amount of time in seconds that it took this DRO to perform a recalculation that resulted in an update to the interest status for this portal.

Last proxy receiver recalc duration

The amount of time in seconds that it took this DRO to perform recalculation that resulted in an update to the status of proxy receivers (create, maintain, or destroy) for this portal.

Other DROs

This display is repeated for each other DRO in this DRO's network.

UM Router Name

The DRO name as assigned via configuration.

UM Router ID

A unique value that the DRO assigns to itself automatically.

Version

A configuration version for the DRO, as seen collectively by the DRO network.

Topology Signature

An identifier for the "map" of this DRO network's routes. This value should be the same for all DROs.

Last Activity n seconds ago

How long since the last time this local DRO received a route info packet from the designated "other" DRO.

Adjacent Domain ID

The configured ID of one of this "other" DRO's connected TRD, plus the cost assigned to the associate endpoint portal. If there are more than one endpoint portals in the DRO, this line is repeated for each.

Adjacent UM Router ID

The automatically assigned ID of one of this "other" DRO's connected DRO, plus the cost assigned to the associate peer portal. If there are more than one peer portals in the DRO, this line is repeated for each.

8.1.5 Path Info

The Path Info page lets you query and display a hop path that messages will take between any two TRDs that you enter into the Domain ID 1 and Domain ID 2 text boxes. Fill in the boxes and click the Calculate Shortest Path button, and you see the following fields:

Hop Count

The number of hops from none node to the next along the displayed route, where a node can be either a DRO or a TRD.

Aggregate Cost

A sum of the cost values of all portals along the displayed path.

Path

A display of the DRO and TRD hops listed in route order from the starting TRD to the ending TRD.

8.2 DRO Log Messages

The DRO daemon generates log messages that are used to monitor its health and operation. You can configure these to be directed to "console" (standard output), "syslog", or a specified log "file", via the <log> configuration element. Normally "console" is only used during testing, as a persistent log file is preferred for production use. The DRO does not over-write log files on startup, but instead appends them.

8.2.1 DRO Rolling Logs

To prevent unbounded disk file growth, the DRO supports rolling log files. When the log file rolls, the file is renamed according to the model:
CONFIGUREDNAME_PID.*DATE*.*SEQNUM*
where:

- *CONFIGUREDNAME* - Root name of log file, as configured by user.
- *PID* - Process ID of the DRO daemon process.
- *DATE* - Date that the log file was rolled, in YYYY-MM-DD format.
- *SEQNUM* - Sequence number, starting at 1 when the process starts, and incrementing each time the log file rolls.

For example: `umrouterlog_9867.2017-08-20.2`

The user can configure when the log file is eligible to roll over by either or both of two criteria: size and frequency. The size criterion is in millions of bytes. The frequency criterion can be daily or hourly. Once one or both criteria are met, the next message written to the log will trigger a roll operation. These criteria are supplied as attributes to the <log> configuration element.

If both criteria are supplied, then the first one to be reached will trigger a roll. For example, consider the setting:

```
<log type="file" size="23" frequency="daily">dro.log</log>
```

Let say that the log file grows at 1 million bytes per hour. At 11:00 pm, the log file will reach 23 million bytes, and will roll. Then, at 12:00 midnight, the log file will roll again, even though it is only 1 million bytes in size.

Note

The rolling logs cannot be configured to automatically overwrite old logs. Thus, the amount of disk space consumed by log files will grow without bound. The user must implement a desired process of archiving or deleting older log files according to the user's preference.

8.2.2 Important DRO Log Messages

Connection Failure Messages

```
peer portal [name] failed to connect to peer at [IP:port] via [interface] [err]: reason
peer portal [name] failed to accept connection (accept) [err]: reason
```

Lost Connection Messages

```
peer portal [name] lost connection to peer at [IP:port] via [interface]
peer portal [name] connection destroyed due to socket failure
peer portal [name] detected dropped inbound connection (read) [err]: reason
peer portal [name] detected dropped inbound connection (zero-len read)
```

Endpoint Messages

If a UMP store is adjacent to the DRO, and the DRO has been restarted, you typically see messages of the form:

```
endpoint portal [name] has no forwarding entry for destination ctxinst [string], dropping msg (lbmc cntl
      ume)
```

These messages are normal, and cease when the DRO has established the forwarding information for the given context.

Peer Messages

```
Acceptor: peer portal [name] received connection from [IP:port]
Initiator: peer portal [name] connected to [IP:port ]
```

8.3 DRO Transport Stats

Using the <monitor> element in a DRO's XML configuration file and the UMS Monitoring feature, you can monitor the transport activity between the DRO and its Topic Resolution Domain. The configuration also provides Context and Event Queue statistics. The statistics output identifies individual portals by name.

Chapter 9

DRO Glossary

Access Control List (ACL)

A portal configuration you can use to filter out messages based on a variety of criteria.

forwarding cost

A value assigned to a portal to help determine best-path routing selection.

UM Router keepalive

Control messages exchanged between DROs to confirm that DROs are still running.

Interest Message

Control messages exchanged between DROs to confirm that DROs are still running.

Originating Transport ID (OTID)

Unique identifier of a message's transport session at the originating source.

portal

A TCP/IP interface (socket) on a DRO through which the DRO passes data. Endpoint portals interface TRDs, and peer portals interface peer portals of other DROs.

Topic Resolution Domain (TRD)

The realm of UDP multicast or unicast connectivity that allows UM topic resolution to occur. Blocking of this UDP connectivity (for example, by a firewall or a restrictive WAN link) defines a TRD's boundaries. Contexts within a TRD must have the same topic resolution configuration option settings (multicast group IP address/port and resolver interface full or CIDR address).

Use Query

A periodic control message distributed to all members of a TRD to verify the continued presence of receivers for a given topic or pattern.

web monitor

A web-based real-time DRO statistics and configuration display.

Chapter 10

Comparison to Pre-6.0 UM Gateway

With the release of Ultra Messaging 6.0, the UM Gateway feature is discontinued and replaced by the Ultra Messaging Dynamic Routing Option (also referred to as the DRO).

The DRO's primary improvement over the UM Gateway is its ability to intelligently select efficient traffic routes from multiple path choices on a dynamic topic-by-topic basis.

Note

This release of the DRO is not backward compatible with earlier versions of the UM Gateway in the sense that you cannot have DROs and UM Gateways in the same network.

10.1 Added Features and Differences

In addition to routing functionality, the following are features of the DRO that were not provided in the UM Gateway:

- Multi-path, ring, or mesh topologies
- Interoperability with MIM and Persistence (see UM Feature Compatibility for complete feature interoperability information)
- Ability to restart the DRO within a transport's activity timeout period
- Reduced topic resolution traffic via more efficient use of Use Queries and Use Query Responses
- The default value for the portal <cost> is 1 (one). 0 (zero) is not a valid cost value.
- The DRO daemon (tnwgd) logs version information on startup.
- Compression and/or encryption may now be applied to peer links.

The following configuration options exist in the DRO but not the UM Gateway. See XML Configuration Reference for more information on these options.

- <name> (as a <daemon> child)
- <route-info>
- <route-recalculation>
- <source-deletion-delay>

- <max-queue>
- <remote-topic-interest>
- <remote-pattern-interest>
- <rate-limit>
- <domain-route>
- <remote-topic>
- <remote-pattern>
- <sourcemap>
- <compression>
- <tls>