



Network Management

Table of contents

UFM Routing Protocols

Configuring Routing Protocol

List of Figures

Figure 0. Image2019 9 19 20 6 5 Version 1 Modificationdate
1716900019777 Api V2

Figure 1. Image2019 6 16 14 58 50 Version 1 Modificationdate
1716900019400 Api V2

UFM achieves maximum performance with latency-critical tasks by implementing traffic isolation, which minimizes cross-application interference by prioritizing traffic to ensure critical applications get the optimal service levels.

UFM Routing Protocols

UFM web UI supports the following routing engines:

- MINHOP – based on the minimum hops to each node where the path length is optimized (i.e., shortest path available).
- UPDN – also based on the minimum hops to each node but it is constrained to ranking rules. Select this algorithm if the subnet is not a pure Fat Tree topology and deadlock may occur due to a credit loops in the subnet.
- DNUP – similar to UPDN, but allows routing in fabrics that have some channel adapter (CA) nodes attached closer to the roots than some switch nodes.
- File-Based (FILE) – The FILE routing engine loads the LFTs from the specified file, with no reaction to real topology changes.
- Fat Tree – an algorithm that optimizes routing for congestion-free "shift" communication pattern.
Select Fat Tree algorithm if a subnet is a symmetrical or almost symmetrical fat-tree. The Fat Tree also optimizes K-ary-N-Trees by handling non-constant K in cases where leafs (CAs) are not fully staffed, and the algorithm also handles any Constant Bisectional Bandwidth (CBB) ratio. As with the UPDN routing algorithm, Fat Tree routing is constrained to ranking rules.
 - Quasi Fat Tree – PQFT routing engine is a closed formula algorithm for two flavors of fat trees
 - Quasi Fat Tree (QFT)
 - Parallel Ports Generalized Fat Tree (PGFT)
PGFT topology may use parallel links between switches at adjacent levels, while QFT uses parallel links between adjacent switches in different sub-trees. The main motivation for that is the need for a topology that is not just optimized for a single large job but also for smaller concurrent jobs.

- Dimension Order Routing (DOR) – based on the Min Hop algorithm, but avoids port equalization, except for redundant links between the same two switches. The DOR algorithm provides deadlock-free routes for hypercubes, when the fabric is cabled as a hypercube and for meshes when cabled as a mesh.
- Torus-2QoS – designed for large-scale 2D/3D torus fabrics. In addition, you can configure Torus-2QoS routing to be *traffic aware*, and thus optimized for neighbor-based traffic.
- Routing Engine Chain (Chain) – an algorithm that allows configuring different routing engines on different parts of the IB fabric.
- Adaptive Routing (AR) – enables the switch to select the output port based on the port's load. This option is not available via UFM Web UI.
 - AR_UPDN
 - AR_FTREE
 - AR_TORUS
 - AR_DOR
- Dragonfly+ (DFP, DPF2)

Configuring Routing Protocol

Network Management tab enables setting the preferred routing protocol supported by the UFM software, as well as routing priority.

To set the desired routing protocol, move one routing protocol or more from the **Available** list to the **Selected** list, and click "Save" in the upper right corner.

Routing Information	
Lid Matrix Dump File	/opt/ufm/files/conf/opensm/lid_matrix.conf
LFTS File	/opt/ufm/files/conf/opensm/lfts.conf
Root Guid File	/opt/ufm/files/conf/opensm/root_guid.conf
Compute Nodes File	N/A
Node IDs File	N/A
Guid Routing Order File	N/A
Active Routing Engine	minhop

The protocol at the top of the list has the highest priority and will be chosen as the **Active Routing Engine**. If the settings for this protocol are not successful, UFM takes the next available protocol.

Routing Information is listed on the top of the screen:

Field/Box	Description
LID Matrix Dump File	File holding the LID matrix dump configuration
LFTS File	File holding the LFT routing configuration
Root GUID File	File holding the root node GUIDS (for fat-tree or Up/Down)
Compute Nodes File	File holding GUIDs of compute nodes for fat-tree routing algorithm
GUID Routing Order File	File holding the routing order GUIDs (for MinHop and Up/Down)
Node IDs File	File holding the node IDs
Active Routing Engine	The current active routing algorithm used by the managing OpenSM

Routing Engine

Select and order the available routing protocol by priority:

Available		Selected
<div><p>Routing Protocol</p><p>Filter... ▾</p><ul style="list-style-type: none">MINHOP <input checked="" type="checkbox"/>UPDNFILEFTREEDORTORUS-2QOSCHAINPQFTAR_UPDNAR_FTREEAR_TORUSAR_DORDFP</div>	<div style="text-align: center;"><p>>></p><p>></p><p><</p><p><<</p></div>	<div><p>Routing Protocol</p><p>Filter... ▾</p><ul style="list-style-type: none">MINHOP</div>

Connect Roots (Leave unchecked if unsure)

© Copyright 2024, NVIDIA. PDF Generated on 06/06/2024