Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 2 October 2025

H. Wang
H. Huang
X. Geng
Huawei
X. Xu
China Mobile
Y. Xia
Tencent

31 March 2025

Adaptive Routing Notification
draft-wh-rtgwg-adaptive-routing-arn-04

Abstract

Large-scale supercomputing and AI data centers utilize multipath to implement load balancing and/or improve transport reliability. Adaptive routing (AR), widely used in direct topologies such as dragonfly, is growing popular in commodity data centers to dynamically adjust routing policies based on path congestion and failures. When congestion or failure occurs, the sensing node can not only apply AR locally but also send the congestion/failure information to other nodes in a timely and accurate manner to enforce AR on other nodes, thus avoiding exacerbating congestion on the reported path. This document specifies Adaptive Routing Notification (ARN), a general mechanism to proactively disseminate congestion detection and congestion elimination information for remote nodes to perform re-routing policies. Particularly for AI workloads like DeepSeek’s MoE models that exhibit dynamic all-to-all communication patterns with bursty traffic characteristics, such mechanisms become crucial to enable immediate network response to transient congestion conflicts.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 2 October 2025.

Table of Contents

1. Introduction

Adaptive routing (AR) is widely used in high-performance computing (HPC) environments with directly connected topologies like Dragonfly[I-D.draft-agt-rtgwg-dragonfly-routing]. These topologies offer advantages such as scalability with small network diameters, making them widely adopted in HPC and supercomputing systems.

In networks with directly connected topologies, multiple non-equivalent paths exist to reach the destination node. Typically, the shortest path is preferred for forwarding traffic. However, traffic congestion can occur on these shortest paths. AR addresses this by enabling nodes to make dynamic routing decisions based on network traffic variations, such as link congestion.

AR is also applicable to symmetrical topologies, which are the most prevalent in current AI data centers, like the Clos topology. In symmetrical topologies, multiple equivalent paths are available. When congestion occurs on one path, AR can adjust traffic flows to avoid the congested path, thus ensuring balanced traffic distribution and optimal path usage.

For example, by proactively detecting link congestion status or receiving remote congestion notifications, network nodes can forward packets along shorter, non-congested paths, improving overall throughput and resilience while reducing latency. When the link is non-congested, packets are forwarded over the shortest paths. When congestion occurs on any shortest path, the local node that detects it applies adaptive routing immediately and advertises congestion signals to other remote nodes. This allows the network to select another non-congested but non-shortest path temporarily until a congestion elimination signal is received. Adaptive routing helps mitigate traffic collisions and utilize idle links, enhancing bandwidth utilization.

When data centers using symmetrical topologies employ Equal-Cost Multi-Path routing (ECMP), AR can correct the membership of ECMP groups by providing timely congestion updates. This ensures traffic is balanced across optimal paths and prevents overload on specific links.

AR mechanisms are also effective in handling path failure scenarios, as path failures can be considered severe congestion cases. The re-routing strategy differs in these cases: when a link failure occurs, no traffic can pass through the failed link, necessitating a complete re-route of all affected traffic. In contrast, when congestion occurs, some traffic can still flow through the link, so AR ensures that only the excess or partial traffic is re-routed, maintaining some level of flow through the congested link.

To standardize the process of disseminating information for triggering re-routing, including but not limited to congestion and failures, the concept of Adaptive Routing Notification (ARN) is introduced. ARN allows for a unified approach to adaptive routing across different network environments, ensuring consistent and efficient handling of network changes and improving overall network

performance and reliability. Additionally, standardizing ARN reduces the need for multiple implementations by switch vendors, simplifying network management and deployment.

This document proposes a proactive notification mechanism for adaptive routing and describes the conditions for triggering dissemination and the information carried in ARN to notify remote nodes for re-routing. ARN can be used for congestion notifications, link failure notifications, and even to convey other relevant network events for re-routing. ARN is applicable to both directly connected topologies and indirectly connected topologies. The detailed mechanisms for detecting congestion or failures are beyond the scope of this document.

1.1. Terminology

AR: Adaptive Routing

ARN: Adaptive Routing Notification

BPT: Best Path Table

ECMP: Equal-Cost Multi-Path routing

HPC: High-Performance Computing

VXLAN: Virtual eXtensible Local Area Network

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. ARN Mechanism

The ARN mechanism primarily consists of three steps:

1. Detect changes in the status of network links/nodes (such as link congestion, link signal interruption, etc.).

2. Assess the impact range of the status change and, if local measures cannot completely mitigate the impact, send an ARN message to specified remote nodes.

3. When remote nodes receive the ARN message, they make rerouting decisions based on the specific information carried by the ARN, thus minimizing the impact on subsequent traffic (e.g., selecting a new path for subsequent traffic).

Here, link congestion is taken as an example to show how ARN works. ARN can be triggered whenever link congestion (e.g., by analyzing the queue length of the output port) is detected to appear or disappear. A congestion signal carried through ARN is sent by the detected node to other nodes of interest (usually the upstream nodes).

Figure 1 depicts a simplified dragonfly topology (only relevant links are drawn). The nodes in each Group are directly connected to each other. The groups are all connected with direct links. As shown in Figure 1, Node1 has a direct link connecting Group1 and Group2. When the direct link (Node1 <-> Group2) is congested, all nodes of Group1 should be notified and immediately update the path selection policy. For example, partial or all flows originating from Group1 to Group2 may choose Group3 as a transmission path instead of using the direct link (Node1 <-> Group2) until congestion elimination.

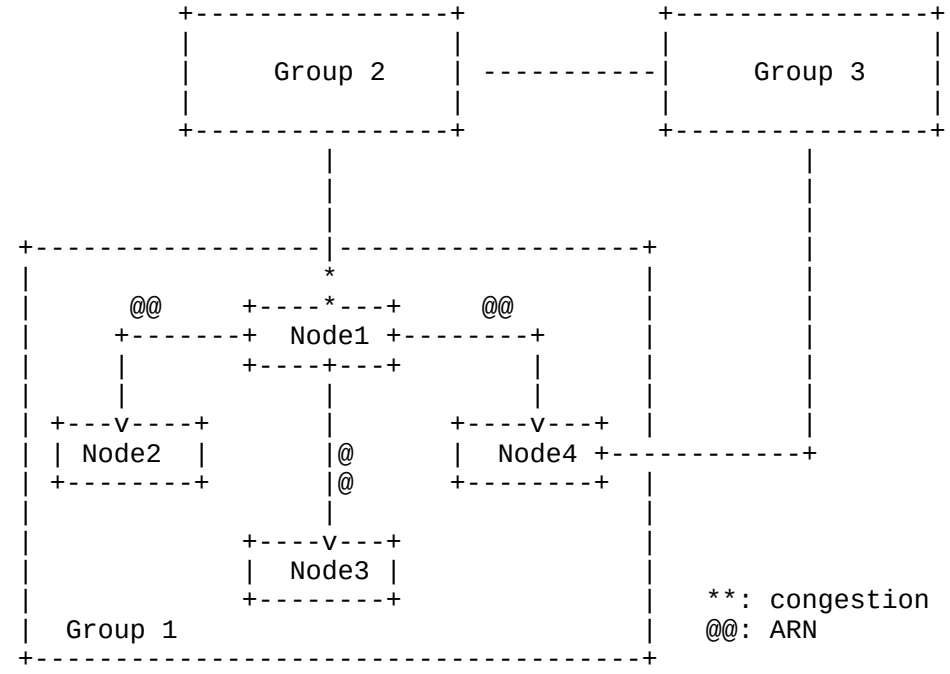


Figure 1: ARN Example in Dragonfly

Figure 2

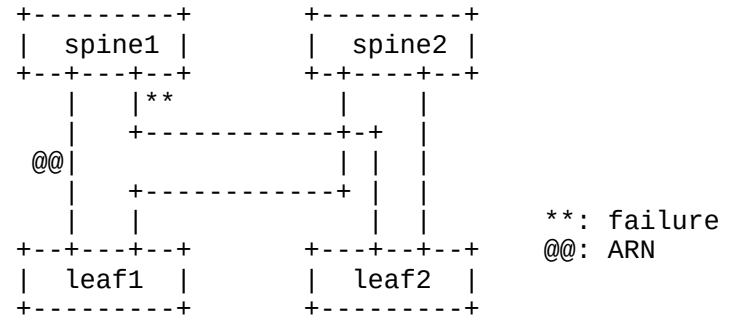


Figure 2: ARN Example in Spine-Leaf

Figure 2 depicts a reduced Spine-Leaf topology with an example of how ARN is triggered in case of a failure. Specifically, when Spine1 detects a failure on the link to Leaf2, and finds no local backup path, Spine1 sends an ARN message to Leaf1, instructing it to reroute subsequent traffic destined for Leaf2 through Spine2.

2.1. Triggering ARN

The local node can sense the change of network states by monitoring interface status, such as bandwidth utilization and queue depth of the interface. The sensing method is out of scope in this document.

When the monitored value exceeds the preset threshold, the state is determined to be congested and a congestion notification is triggered. When the monitored value falls back below the preset threshold, the state is determined to be non-congested and a notification of congestion elimination is triggered.

When the local node detects any change in congestion status, it can send the corresponding ARN continuously to other network nodes in the same group. The notifications can be sent to multiple nodes using multicast technology provided by the network. ARN packets SHOULD be set as high priority to ensure timely processing. The congestion level is RECOMMENDED to be included in ARN for fine-grained control of adaptive routing.

The local node can sense the change of network states by monitoring interface status, such as bandwidth utilization and queue depth of the interface. The methods for detection and sensing can vary widely, including techniques such as active probing, passive monitoring, and others. However, the specific methods are out of scope in this document.

However, the detection of a state change does not necessarily trigger the ARN mechanism. Nodes can decide whether to trigger remote notifications based on predefined rules. For instance, local measures might be sufficient to handle the issue. If a link failure occurs but the local node has multiple backup links available, the local rerouting might suffice to resolve the problem without needing to trigger an ARN.

When the local node decides to trigger ARN based on the change in specific network status, it can send the corresponding ARN continuously to other network nodes. The ARNs could be sent by unicast or multicast. ARN packets SHOULD be set as high priority to ensure timely processing.

2.2. Receiving ARN

After receiving an ARN, the node generally performs re-route operations, which include but not limited to:

* Selecting a new optimal path for the traffic that avoids the congested or failed link.

* Adjusting the sending rate of traffic to prevent overload and reduce congestion.

If the node determines that it cannot effectively re-route the traffic based on the received ARN, it may propagate the ARN information to other nodes in the network, continuing the dissemination process to ensure network-wide adaptation and optimal traffic flow.

3. Adaptive Routing Notification

3.1. Basic Concept

An ARN packet should include two kinds of information:

* Information reflecting the type of notification and quantifiable metrics (e.g., congestion level). The Metric value helps in quantifying the severity of the congestion or failure, enabling fine-grained control of adaptive routing.

* Information carrying details about the affected object (e.g., affected traffic, affected paths), for example, router identifier connected by the compromised link or identifiers of flows that are impacted by the congestion or failure.

These details are essential to assist remote nodes in making informed rerouting decisions, ensuring minimal disruption and optimal network performance despite the presence of congestion or failures.

Whenever a network node receives an ARN packet indicating congestion detection, for example, it would evaluate the optimal forwarding path in its local best path table (BPT). If the optimal path passes through the affected interface, the network node deletes this path from the BPT and selects other sub-optimal paths. How to respond to ARN packets is typically related to the specific device's rerouting implementation mechanism for AR.

ARN can also be used to notify the elimination of specific network conditions (e.g., congestion recovery). When such an ARN message is received, the previously made rerouting decisions can be revoked. In this case, each ARN message should be configured with an identifier (carried through parameters) to ensure the correspondence between the state notification and the state revocation notification. If ARN is not used for elimination, mechanisms such as timeouts can be employed to revoke rerouting decisions.

Simple and direct ARN messages may cause routing oscillation issues and packet reordering problems within the same flow. These issues can be better addressed in future enhancements. Additionally, ARN is primarily a rapid rerouting mechanism and is typically used in conjunction with robust BGP mechanisms. Once BGP routes converge, they will replace the rerouting strategies triggered by ARN, ensuring routing correctness, loop-freeness, and reducing the side effects caused by the simplistic ARN mechanism.

3.2. Packet Format

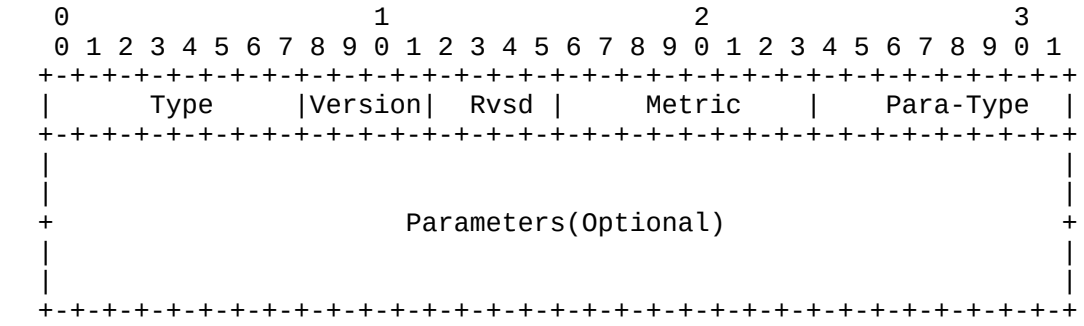


Figure 3: ARN Format

where:

Type: This field indicates the purposes of ARN. Type 1 indicates this notification is for notifying congestion detection remotely to trigger adaptive routing. Type 2 indicates this notification is for notifying congestion elimination remotely to revoke adaptive routing. Type 3 indicates this notification is for notifying failure detection remotely to trigger adaptive routing. Type 4 indicates this notification is for notifying failure elimination remotely to revoke adaptive routing.

Version: This field indicates the version number. The default value is 0.

Rvsd: Reserved.

Metric: Quantified value. For example, it can be used to notify the degree of congestion or indicate the variation in available bandwidth.

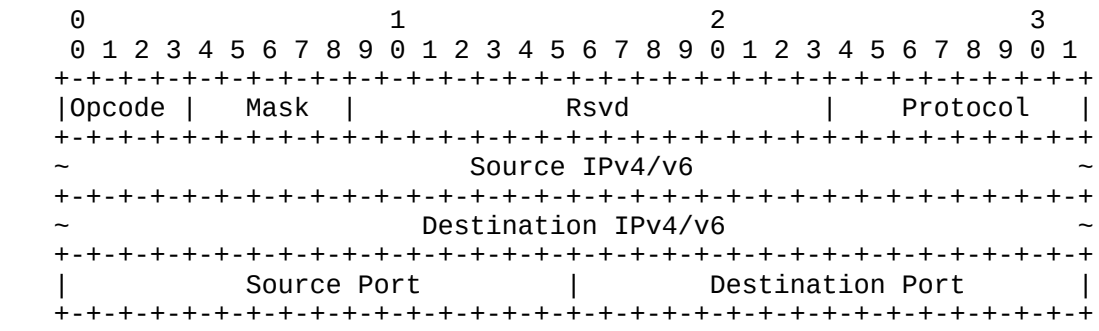
Para-Type: The Para-Type field is an 8-bit bitmap that specifies which parameters are included in the Parameters field of the ARN packet. Each bit in this field corresponds to a specific parameter. When a bit is set to 1, it indicates the presence of the corresponding parameter. The following subsections detail the explanation of each bit in the Para-Type field.

Parameters: The parameters field can carry the information of affected object to help other devices determine the target of adaptive routing. The presence of parameters is indicated by the Para-Type bitmap. The packing order of the parameters follows the bit order specified in the Para-Type bitmap field.

3.2.1. Illustration of Para-Type and Corresponding Parameter

3.2.1.1. Para-Type Bit 0

When bit0 of Para-Type is 1, the following parameter is concluded in Parameters to indicate the identifier of affected flow (five-tuple from packet header):



where:

Opcode: This field indicates either IPv4 address or IPv6 address is used in the parameter.

Rsvd: Reserved for future use.

Mask: A bitmap used to indicate the presence of the subsequent 5 fields, excluding the reserved field. Each bit in this field corresponds to a specific domain, with a value of 1 indicating the presence of the domain and 0 indicating its absence.

Here’s the breakdown of each bit in the Mask field:

* Bit 0 (Protocol): Indicates whether the Protocol field is present.

* Bit 1 (Source IPv4/v6): Indicates whether the Source IP address field is present.

* Bit 2 (Destination IPv4/v6): Indicates whether the Destination IP address field is present.

* Bit 3 (Source Port): Indicates whether the Source Port field is present.

* Bit 4 (Destination Port): Indicates whether the Destination Port field is present.

Protocol: Indicates the specific protocol type used by the packet, such as TCP or UDP.

Source IPv4/v6: : The IP address of the sender, which can be in IPv4 or IPv6 format determined by Opcode.

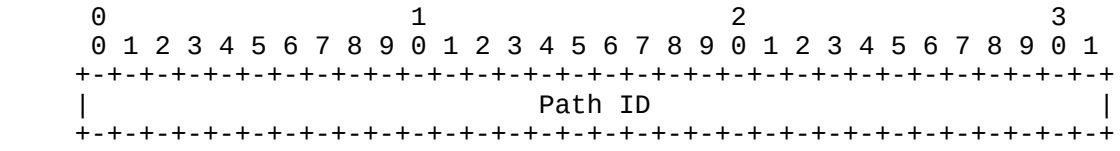
Destination IPv4/v6: : The IP address of the receiver, which can be in IPv4 or IPv6 format determined by Opcode.

Source Port: : The port number used by the sender.

Destination Port: : The port number used by the receiver.

3.2.1.2. Para-Type Bit 1

When bit1 of Para-Type is 1, the following parameter is concluded in Parameters to indicate the identifier of affected path:



Path ID: The 32-bit field is used to uniquely identify the affected path in the network.

3.2.1.3. Para-Type Bit 2 to Bit 7

Bits 2-7: Reserved for future use or other parameter types.

In scenarios such as VXLAN tunnels, there may be both inner and outer five-tuple flow identifiers. Different parameters can be used to distinguish between these two types of identifiers, allowing for more granular routing control. Specifically:

* Bit 0 is used for the outer five-tuple identifier (related to the VXLAN tunnel).

* Additional bits (e.g., Bit 2) can be used for the inner five-tuple identifier (related to the actual payload traffic within the tunnel).

By using different bits in the Para-Type field, the ARN mechanism can indicate the presence of these different parameters, enabling precise and fine-grained adaptive routing decisions.

4. Security Considerations

TBD.

5. IANA Considerations

TBD.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

6.2. Informative References

[I-D.draft-agt-rtgwg-dragonfly-routing] Afanasiev, D., Glebov, R., and J. Tantsura, "Routing in Dragonfly+ Topologies", Work in Progress, Internet-Draft, draft-agt-rtgwg-dragonfly-routing-01, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-agt-rtgwg-dragonfly-routing-01>.

Acknowledgements

Contributors

Authors' Addresses

Haibo Wang
Huawei
Email: rainsword.wang@huawei.com

Hongyi Huang
Huawei
Email: hongyi.huang@huawei.com

Xuesong Geng
Huawei
Email: gengxuesong@huawei.com

Xiaohu Xu
China Mobile
Email: xuxiaohu_ietf@hotmail.com

Yinben Xia
Tencent

Email: forestxia@tencent.com