

Study of Rifi and Rift Protocol in Data Center Network

Pooja Kaplesh*

Department of Computer Science, Chandigarh University, Mohali, India

Review Article

Received: 06-Jun-2022,
Manuscript No. GRCS-22-66038;
Editor assigned: 9-Jun-2022, Pre
QC No. GRCS-22-66038 (PQ);
Reviewed: 26-Jun-2022, QC No.
GRCS-22-66038; **Revised:** 07-
Aug-2022, Manuscript No. GRCS-
22-66038 (R); **Published:** 15-
Aug-2022, DOI: 10.4172/2229-
371X.13.4.007

*For Correspondence : Pooja
Kaplesh, Department of
Computer Science, Chandigarh
University, Mohali, India, Tel:
7018389065;
Email: poojakaplesh4@gmail.com

Keywords: Data centre network; RIFL;
RIFT; Protocol; Frame structure

ABSTRACT

Delay sensitive applications are increasingly finding their way into data centers. The network interface's higher latency can limit the performance of such applications. Because the traditional network stack built for both LAN and WAN, includes a significant amount of redundancy which is not necessary in data center networks. This paper presents the study of high scale network protocols named RIFL and RIFT that can replace traditional network stacks and meet the exact needs of data center network communications. Comparative features and benefits of these protocols in high scale network system are presented. Frame and header structure for both the protocols are also discussed.

INTRODUCTION

The standard TCP/IP stack is designed to function satisfactorily in both a Local Area Network (LAN) and a Wide Area Network (WAN) (WAN). The physical properties of a LAN and a WAN are drastically different. TCP/IP and User Datagram Protocol (UDP)/IP have too much redundancy in terms of capacity and latency when used in a LAN. Because the diameter of a data center server room is rarely greater than 100 meters, a DCN is effectively a LAN [1]. There should be a more efficient protocol stack that fits the specific requirements of a DCN. RIFT (Routing in Fat Tree) was built from the ground up to revolutionize data center routing. It comes with extensive capabilities and self-optimizations, and it fills in the gaps left by other routing protocols. For enterprise to hyper scale data centers, RIFT might be an autonomous all in one routing solution [2]. RIFT is finalized as a supporting and recognized standard to the standards community.

LITERATURE REVIEW

RIFT is a zero operations cost protocol for routing to route packets in topologies based on CLOS and fat tree networks. It is a mix of distance vector and link state approaches that has various advantages for IP fabrics, including simplicity of management and increased network robustness. RIFT routing protocol is an open and free standard. It's a cross between a distance vector protocol for the leaves that employs diffused computation and a link state algorithm for the spines that uses distributed flooding and computation [3]. To put it in another way, when the RIFT protocol is activated, devices broadcast their link-state information to the north side, whereas every switch except the leaf generates a default route that is flooded in the southern direction (under normal conditions). With the rising deployment of IP forwarding-based data centres, Interior Gateway Protocols (IGPs) and BGP are being used to control the critical routing decisions in CLOS and fat tree architectures (also known as the spine and leaf model). The methodology of these protocols is based on difficult and expensive operational extensions that fall short of the requirements of IP fabrics [4]. This is because the IGP and BGP protocols were developed for general and sparse

network topologies. Routing In Fat Trees (RIFT) solves these issues and adapts to changing IP infrastructure needs. Following are the essential aspects of Routing In Fat Trees (RIFT) protocol:

- Topologies with fat trees are automatically constructed.
- Reduces the quantity of flooding automatically.
- Reduces the quantity of routing state information stored at each data center network level.
- Based on available bandwidth, automatically rebalances traffic toward the spine.
- To avoid black holing and poor routing, prefixes are automatically disaggregated when links and nodes fail.
- After protocol convergence, synchronizes a tiny key-value data store that can be used to bootstrap higher levels of functionality on nodes.

RIFL is a multi lane protocol that can scale from 500 Mbps to over 200 Gbps in speed. It works with the majority of modern FPGAs. It has the potential to allow data center networks with low latency, high throughput, flexibility, scalability, and lossless performance. The frames are used by RIFL [5].

- **RIFL TX and RX protocols:** TX logic works in six stages in it, send pause, pause, normal, retrains and send retrains. **RIFL RX logic works in five stages:** Re-transmit request out-of-sync, pause request, flow control and frame error.
- **Re transmission:** It operates in three modes: no errors in either direction, errors in one of the directions and errors in both directions.
- **Flow control:** To maintain flow control, a buffer is inserted between the RX logic and the user interface.
- **Clock compensation:** The faster endpoint's TX logic can proactively manage its rate.
- **Channel bonding:** Channel bonding must be used to combine the bandwidths of numerous transceivers in order to attain hundreds of gigabytes per second bandwidth.

Data center network protocols

RIFL frame structure:

- **Header field:** The data frame header must have the following information: Frame ID, checksum, payload count of valid bytes, line code header and end-of-packet marker.
- Data frames fields (Table 1).

Table 1. Data frame structure.

Frame size	Syncword (SYN)	Payload	Meta code*	Format code	Verification code
Fixed frame size	2-bit line code header	up to 2048 bit	2-bit,	8-bit field	Max (Frame ID, checksum)

SYN: This is a 2-bit line code header. It's also used to indicate if a frame is a data frame or a control frame. SYNs in data frames are set to 2'b01 in Verilog constant notation, while SYNs in control frames are set to 2'b10.

Payload: User payload is limited to 128, 256, 512, 1024, and 2048 bits. Both the data frame and the verification must be compact in order to minimize latency and maximize bandwidth efficiency [6]. Because the data frame must be a power of two and no less than 128, and the frame code can only support up to 2048 bits of payload, the data frame options are 128, 256, 512, 1024, and 2048.

Formatting code: The formatting code is a field of 8 bits. When the Meta code shows that not all bytes in the payload are valid, it is utilized to indicate how many bytes are legitimate.

Verification code: Let size of frame ID signifies the size of the frame ID field and size of checksum signify the size of the checksum. Verification code is calculated as Exclusive-OR (XOR) of frame ID and the checksum. Therefore verification= maximum of (Size of frame ID, size of checksum).

Meta code: On the basis of code value, check if payload partially valid, all bytes of the payload is valid and not valid (Tables 2 and 3).

Table 2. The table shows the meta code and payload validity.

Meta code	Payload validity
0	Invalid
1	Valid
10	Valid
11	Valid

- Control frame

Table 3. Control Frame structure.

Control code	Syncword (SYN)	Verification code	Reserved
16-bit	2-bit	Max (Frame ID, checksum)	Size of data frame size of verification code control code SYN

DISCUSSION

In control frames, the SYN and verification code perform the same functions as in data frames. The following are the control codes:

Idle: This signal indicates that the sender is not transmitting data normally. When the sender is in the transition between the pause, re transmit, and regular states, this code is sent out. The following part will go through each state in further detail [7].

Pause request: When the link is out of sync, the receiver sends this code. It informs the sender to take a break from sending data.

Request to retransmit: When a faulty verification code is encountered, the receiver sends this code. It instructs the sender to change from regular mode to emergency mode.

RIFT protocol

The Routing in Fat Trees (RIFT) protocol uses a combination of link state and distance vector techniques to satisfy the needs of routing in fat tree networks, with link-state pointing to the spine and distance vector pointing to the leaves. RIFT focuses on networks with regular topologies, high connectedness, defined directionality, and huge scale using this hybrid technique."

- **RIFT Structure and operations:** RIFT structure defines various operations of RIFT protocol [8]. "It's critical that nodes participating in the protocol require very little configuration and can join a network as leaf nodes simply by joining to the network with the default configuration." IPv6 and IPv4 should both be supported by the protocol."
- **Neighbor discovery:** Through the exchange of Link Information Elements (LIE), RIFT automatically discovers neighbors, negotiates Zero Touch Provisioning (ZTP), and detects any mis-cablings. LIE messages are encrypted in an "envelope" that helps with authentication and security. LIEs are always sent with a TTL (IPv4) value or HL (IPv6) value 1 in order to prevent reaching beyond a single layer 3 hop (Table 4).

Table 4. Default RIFT multicast addresses.

Address family	Default multicast address
IPv4	224.0.0.120
IPv6	FF01:A1F7

• Header Fields

- Local ID: Local ID of the link
- PoD: Local node's PoD value
- MTU: Layer 3 MTU of the local link, which is used to discover MTU mismatches
- Neighbor: Used to "reflect" a neighboring nodes system ID and link ID

• RIFT States

- One way: Initial state.
- Two way: The distant node has sent a valid LIE to the local node.
- Three way: The remote node sees the local node's System ID in the LIE. Link IDs must also match in circumstances where parallel links are employed.
- Multiple neighbors wait: Local node sees multiple neighbors on a single link and initiates a hold down timer before processing LIEs (Table 5).

Comparative features

Table 5. Features of RIFT and RIFL protocols.

Features	RIFL	RIFT
Design	Link layer design	Network layer design

Type of protocol	Multi lane protocol	Zero OpEx routing protocol
Advanced version	Variant of TCP/IP Protocol network.	CLOS based and fat tree network topologies variants
Hybrid protocol	First layer in three-layer protocol stack	A protocol that combines the link-state and distance vector protocols.
Header field	Frame ID, checksum, number of valid bytes in the payload, end of packet marker, and line code header are all included.	Contains local ID, PoD, MTU of the local link and Neighbor
Optimality	Lossless point to point optimum links with ultra low latency and high band width are provided.	Designed to maximize the use of multi-core processing architectures.

CONCLUSION

The protocols discussed in this paper are very useful for communication in hyper scale data center network. RIFL is having low delay or latency and is a dependable link layer protocol in network three layer stack that we have presented. RIFL can provide lossless point-to-point networks with very low latency and a large bandwidth due to its revolutionary in band re transmission algorithm. On the other side, RIFT is a routing protocol designed for densely nested topologies like fat tree. Various features and frame structure of both protocols are presented in this paper.

REFERENCES

1. Qianfeng Shen, et al. "RIFL: a reliable link layer network protocol for data center communication". J Opt Commun Net. 2022;14:111-126.
2. Aelmans M, et al. Day One: Routing in Fat Trees (RIFT). Juniper Networks, USA, 2020;1-237.
3. Przygienda T, et al. RIFT: Routing in Fat Trees. Internet Draft Draft Ietf Rift Rift-12, IETF Secretariat. 2020;1-135.
4. Correa RS, et al. "Ultra low latency communication channels for FPGA based HPC cluster." Integ. 2018;63:41-55.
5. Alizadeh M, et al. "Data center TCP (DCTCP)." SIGCOMM Comput. 2010;63-74.
6. Bari MF, et al. Data Center Network Virtualization: A Survey. IEEE Commun Surv Tutorials. 2013;15:909-928.
7. Li D, et al. On the design and analysis of data center network architectures for interconnecting dual-port servers. In: INFOCOM, 2014 Proceedings IEEE, Toronto, Canada. 2014;1851-1859.
8. Chen K, et al. Survey on routing in data centers: insights and future directions. IEEE Netw. Evanston, USA, 2011;25:6-10.