Advanced Computer and Communications Networks:

Analysis of Bidirectional Forwarding Detection (BFD) on Link Aggregate Groups (LAG)

Presented by:
Josue Contreras and Sharafuddeen Nalakath

Dr. Richard A Stanley
Worcester Polytechnic Institute
December 2020

Abstract

While the current implementation of Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces has been standardized, there existed a divide between these two networking technologies that pushed for standardization by the Internet Engineering Task Force (IEFT). In this paper, we take a look at the development of the BFD standard, its importance in networking, its integration with LAG, and its challenges. We also review current implementations of BFD on LAG networks for fault detection.

Introduction

Computer networks are prevalent in homes, schools, offices, and modern industries around the world. The everyday use of information has created a dependency of these networks. Nowadays a stable and consistent connection is expected by many network users. Therefore networks must support higher traffic rates and adapt to topological changes seamlessly. More devices means more bandwidth on the physical layer. Theoretical and historical data in the last 36 years proves that users' bandwidth grows by 50% per year according to Neilsen's Law [1]. To support such increasing bandwidth demands the Link Aggregation Group method was created and standardized in 2000. The LAG standard allowed manufacturers to produce compatible products to be used in networks today that support such high data rates, high amounts of traffic, and multiple users.

Networks are built of many components and can suffer from component failure. Network failure can be caused at any layer or within any component in a network. Therefore, networks must be able to adapt quickly and redirect traffic. Some network protocols like OSPF are able to detect network failures and allow routers to redirect traffic through other links. The fault detection done in such a protocol is slow and prone to traffic loss due to the non-aggressive timer timeouts. These protocols also have high overhead and are not capable of responding quickly in the high data rate networks of today. This poses a challenge, as link failure and a slow detection can reduce a network's reliability. This is even more pressing when it comes to networks using LAGs as the higher bandwidth virtual channel may suffer more traffic loss than previous single physical member links that connected adjacent nodes.

For these reasons Bidirectional Forwarding Detection was created to meet this demand of faster fault detection for no packet loss. In 2011 the BFD protocol was standardized under RFC 5880. This networking protocol was designed to be lightweight and provide failure detection between two forwarding engines. Specifications of this protocol covered only adjacent nodes connected through a single link network as intended. Vendors started implementing it themselves in their own mechanism. The caveat with the first introduction of this protocol was that there were no specifics on how BFD should be implemented on LAGs. This left vendors to implement exclusive solutions that were not cross-compatible and didn't use the core purpose of BFD at its fullest.

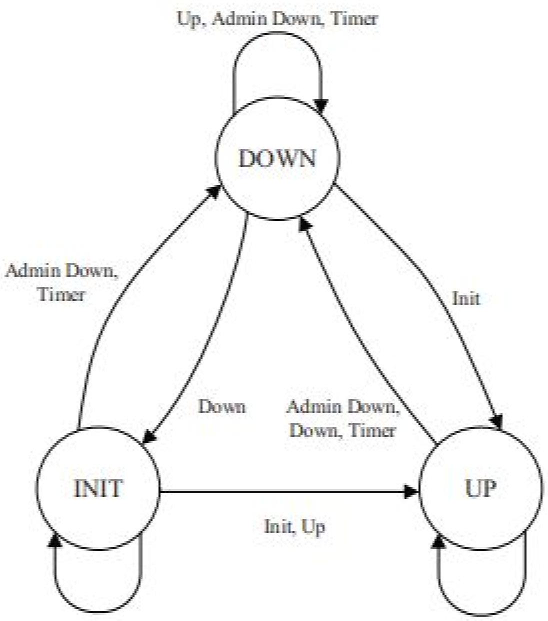
Here we will introduce why the core purpose of BFD was not used by the vendor's implementations. BFD is a protocol that can be used on any layer and as stated previously it was designed on the assumption that there is usually one link that a packet can take. For example, in LAG there are multiple member links that the packet can take and BFD would have to choose which one to take. Therefore the following challenge arose, without BFD knowing the internal knowledge of LAG, the fast detection from BFD would be useless. This was because of the slow failure detection time that LAG had with the Link Aggregation Control Protocol (LACP) compared to the BFD timers. This also meant that implementations from vendors had many issues that defeated the purpose of BFD to detect faults fast. The design of these implementations handed over the failure detection to higher level protocols like LACP, as a result packets/traffic got lost. Therefore IEFT standardized BFD on LAG (RFC 7130) using micro sessions on each LAG member, this results in zero traffic loss and a common detection mechanism [2].

Analysis of BFD and/on LAG

Bidirectional Forwarding Detection (BFD) [3, 4]

BFD provides a fast failure detection mechanism for various types of links. BFD mechanism works in networks consisting of point-to-point links, irrespective of data protocol used, either over network or link layer. This mechanism is typically used in networks that have high bandwidth and requires fast failure detection.

The state machine describing a BFD session is given below:



BFD State Machine [3]

The states of a BFD session are described below:

ADMIN-DOWN: The session transitions to this state when the administrator manually disables the interface. The interface does not respond to or send any control packets in this state. As the behavior of the system in this session is the same as DOWN state, the ADMIN-DOWN state is not represented as a separate state in the state machine.

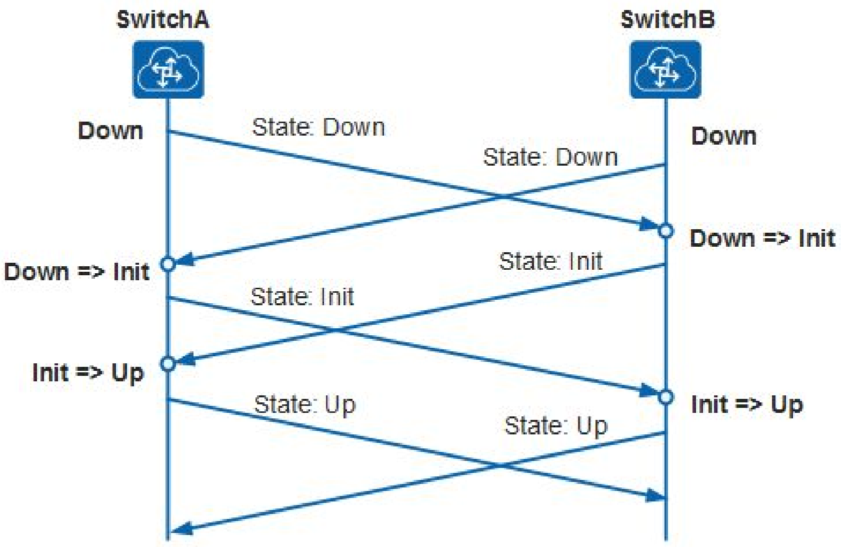
DOWN: A BFD session is in this state upon initialization. The session transitions to the next state upon receiving a BFD message from the neighboring BFD process. If the remote BFD process sends a message that it is also in DOWN state, then the state transitions to INIT. If the neighboring process indicates INIT state, then the state transitions to UP state.

INIT: A BFD that is in DOWN state transitions to INIT when it receives a control packet indicating that the remote BFD process is in DOWN state. When the local BFD session is in INIT state, it transitions to UP state when it receives a control packet indicating that the remote BFD process is in INIT or UP state. If the local process expires the detection time or the remote process goes to ADMIN-DOWN state, the local process transitions to DOWN state.

UP: In this state, both local and remote BFD processes are in the UP state. Local BFD process transitions to this state when it is in DOWN state and remote process indicates it is in INIT state, or if the local process is in INIT state and the remote process indicates it is in INIT or UP state.

BFD works with other protocols such as OSPF and LAG. These protocols notify BFD to establish sessions. The BFD interval is negotiated during session establishment. If a node does not receive any BFD control packets within this interval, a failure notification is sent by BFD to the associated protocol. The associated protocol handles the failure notification appropriately.

Prior to session establishment, BFD is in either active or passive mode. In active mode, a BFD process sends BFD control packets irrespective of whether it receives any BFD control packets from a remote BFD process. In passive mode, a BFD process does not send any BFD control packets until it receives a BFD control packet from its peer. At least one end of the two processes should be in active mode to establish a session. The following figure shows an example of BFD session establishment between two network switches.

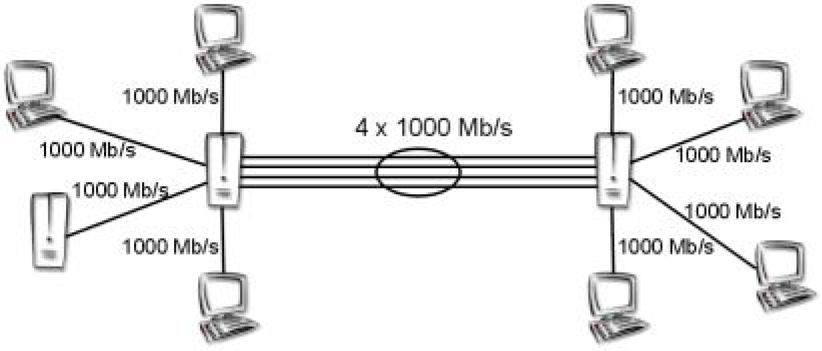


Each BFD session has a transmit timer interval that is negotiated between peers. Before this negotiation, BFD control packets are sent once per second.

After establishment of the BFD session and negotiation of transmit timer interval, both processes begin transmission of BFD control packets at the negotiated timer interval. The detection timer is reset upon reception of a BFD control packet from the peer process. If the BFD control packet is not received within the timer interval, the BFD state transitions to DOWN state. This state transition is notified by the BFD process to the associated protocol service. The protocol service takes appropriate measures upon reception of the state transition notification.

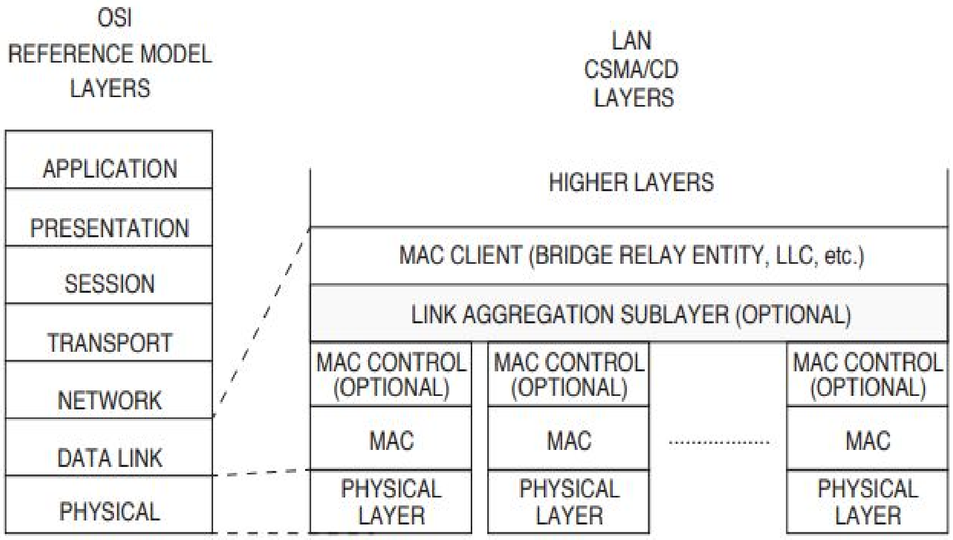
Link Aggregate Group (LAG)

LAG, defined in IEEE 802.1AX [5], provides mechanisms to combine multiple physical links into a single logical link. The bundling of multiple physical links, also called member links, results in an increase of bandwidth. Better connectivity is developed as the redundancy of member links allows capacity to be redirected to the remaining operational member links of the LAG. LAG also adds resilience to the network. For example, if a member link fails within a LAG it will not affect higher level protocols as this was designed to failover completely seamlessly. Failover in networking means the act of switching to a redundant component automatically. Link Aggregation also provides load balancing by distributing the processing and communication activities across several links in a Link Aggregation Group so that no single link is overwhelmed. The following figure shows two servers interconnected by an aggregation of four 1000 Mbps links [6].



LAG pertains to the aggregation or building of physical links to create a higher bandwidth medium. How traffic is distributed through the individual member links and abstracted, to make this aggregation seem as a single logical link, is done by protocols like Link Aggregation Control Protocol (LACP). It is defined in IEEE 802.3ad as the protocol that is required for dynamically exchanging configuration information among systems within the Link Aggregation Group. LACP enables cooperating systems to automatically detect the presence and capabilities of each other. Using LACP, it is possible to specify which links in the system can be aggregated to form the Link Aggregation Group. LACP also enables failure detection of links in a LAG. However, it has some limitations compared to BFD: (1) failure detection takes comparatively more time (2) the protocol is relatively more complex (3) It is a Layer 2 (L2) protocol. Hence, it won't be able to detect failures in Layer 3 (L3).

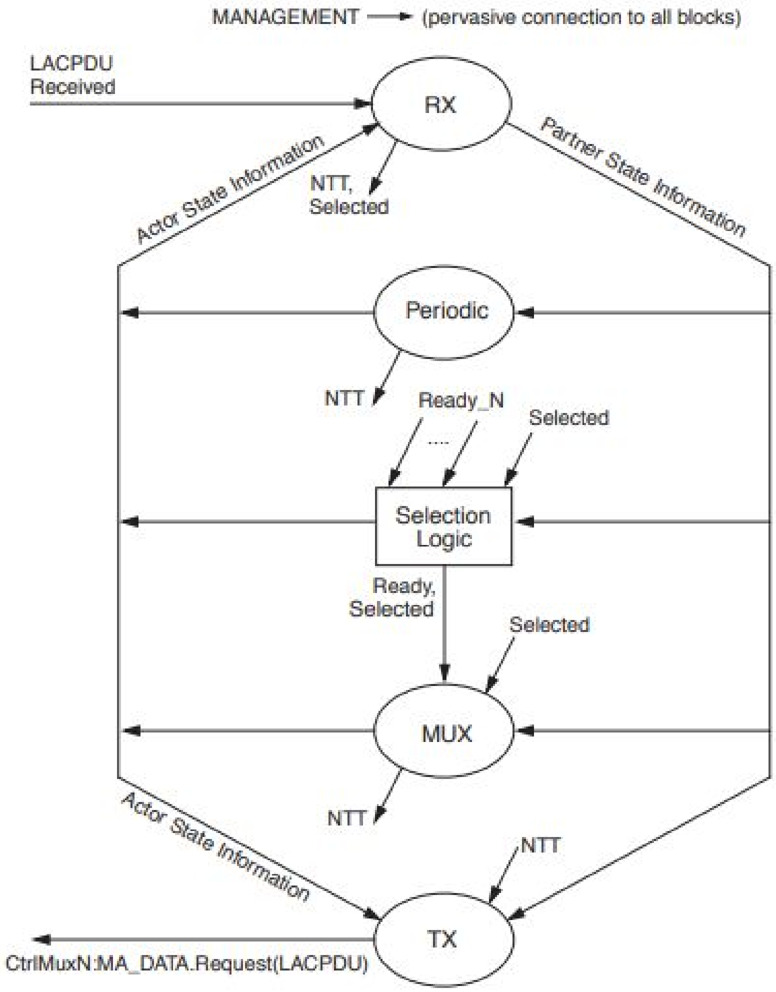
The following diagram shows the architectural positioning of the Link Aggregation sublayer, per LACP. As evident from this diagram, LACP is a data link layer protocol (L2).



The operation of LACP is governed by multiple state machines that work together. Each of these state machines perform a distinct function. These state machines are:

a) Receive machine: Receives LACPDUs from the partner, records the information contained, and times it out based on a preconfigured timeout value.
b) Periodic transmission machine: Determines whether two nodes will exchange LACPDUs periodically to maintain an aggregation.
c) Selection logic: Selects the Aggregator to be associated with a port.
d) Mux machine: Handles attaching and detaching of a port from its aggregator. Also turns off or on collection and distribution at ports as required by the current protocol information.
e) Transmit machine: Responsible for handling transmission of LACPDUs, both on demand from other state machines or on a periodic basis.

The need to transmit an LACPDU is signalled to the transmit machine by asserting the Need-To-Transmit (NTT) identifier. Interrelationships among state machines of LACP are shown in the following diagram.



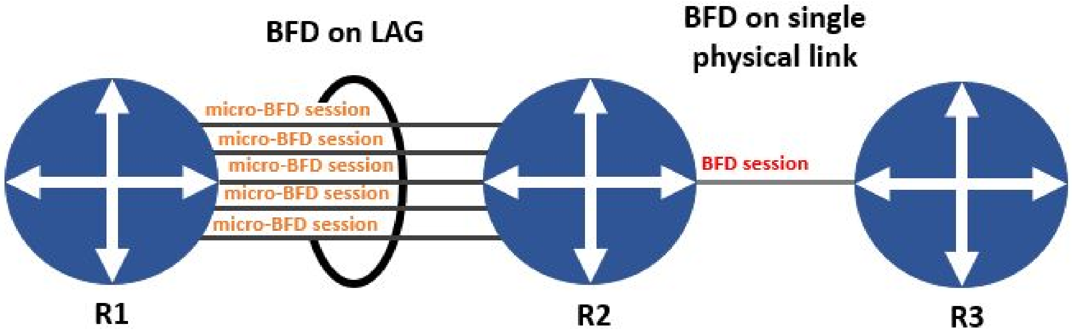
As evident from the above diagram, LACP is a complex protocol.

BFD on LAG [7]

In this section we analyze the BFD on LAG interfaces standard RFC 7130 provided by the IEFT. As seen in the previous section LAGs provide higher throughput beyond what a single connection could offer and member link redundancy allows for better resiliency in networks. Additionally, the LACP method is usually used for LAGs that provide additional functionality to form a logical virtual channel out of the bundled physical links. Although the LACP provides the LAGs with failure detection on a per physical member link, in today's networks, it takes comparatively more time caused by the overhead on the protocol. Therefore before the establishment of RFC 7130, protocols like LACP gave BFD an altruistic behavior since BFD supplies information to other protocols which decide how to react to the information themselves. As a result these implementations did not take advantage of the fast functionality of BFD. Now with the RFC 7130, BFD is able to be implemented on top of any aggregation protocol like

LACP on LAG while still providing demanding failure detection times. Moreover BFD enables operators who have already deployed mechanisms to use a common failure detection method.

To implement BFD on LAGs the RFC 7130 standard states that each LAG member link has to run a separate Asynchronous mode BFD session, defined as a micro-BFD session. It is important to note that LAG member links are treated as one big interface by routing protocols (L3 protocols) like OSPF and BFD is done on individual link members of the LAG, these are the micro-BFD sessions. Given the altruistic behavior of BFD, signaling of these micro-BFD sessions to router protocols may be done on the load-balancing table and managed by the interface to decide on possible termination of the LAG component. The BFD negotiation procedure is the same as the one described for a single link on RFC 5880. The distinction from a single physical link BFD session to BFD on LAG is that each individual micro-BFD session has its own unique local discriminator values, maintains its own state variables, and has its own independent state machines on each member link. In conjunction when BFD is used on LAGs, the destination UDP port is 6784 for BDF packets. This is an important specification for BFD on LAGs compared to BFD over single-hop IP. The following diagram shows BFD on LAG with each member link running a separate mico-BFD session between R1 and R2. It also shows a BFD session on a single physical link between R2 and R3 to show the distinction.



BFD plays a crucial role as it dictates the connection between two adjacent nodes connected by a LAG. This can be seen in the interaction between LAG and BFD. Micro-BFD sessions must be active when its respective member link is active. This means that the LAG member link has to be in Distributing or Standby states. If the LAG member link is inactive the micro-BFD session is turned off. For the interaction between BFD and the load-balancing algorithm there are two rules to follow. The first one dictates that even if a protocol like LACP considers a link ready for traffic forwarding it cannot forward until all micro-BFD sessions on that individual LAG member link are in the Up state. The second rule states that in the case of separate load-balancing tables on member link, the algorithm can enable the member link of the address family with its respective table. This means that if IPv4 and IPv6 micro-BFD sessions exist on a member link then they can be separately enabled on the link by virtue of using micro-BFD sessions.

An important aspect that is overlooked when implementing two technologies like BFD on LAG Interfaces is security. By introducing a component in a system, like BFD into LAGs, vulnerabilities might be introduced. This is a consideration to take into account as the network might be able to handle slower failure detection times rather than introducing vulnerabilities to the system. In comparison the network might prioritize failure detection over security as well and mitigate security by other methods. In the case of BFD on LAG, RFC 7130 states that the security concerns from the standardized BFD in RFC 5880 can be inherited. Security regarding failure detection methods is important as the network is managed and adapts accordingly. As stated in RFC 5580, attacks on BFD sessions may be very serious since they are typically used to determine the stability of the network. An attack could declare an erroneous state and create a Denial of Service (DOS) attack. This introduces a serious vulnerability that has to be taken into consideration more so because BFD doesn't prevent a DOS attack. Luckily, other methods have been created to add security. In the case of BFD, there exists two possible mechanisms that can mitigate such attacks, Time to Live (TLL)/Hop Count and enable the authentication section on the BFD packet. These security methods protect BFD from attackers that might want to introduce erroneous BFD control packets in the network.

BFD has great adaptability in networking systems as it is able to be applied on any networking layer and in tandem with other protocols. This enables micro-BFD sessions on LAGs to use IPv4 and IPv6 addressing using IP/UDP encapsulation. This adds flexibility to member links as it allows them to have two micro-BFD sessions running IPv4 and IPv6. The only caveat when doing this on LAG member links is that when a member link uses an address family, all other member links of the LAG must use the same address family.

The BFD on LAG standard RFC 7130 allows for the correct implementation and capilatizes on BFDs fast failure detection. As seen in this section BFD allows for a better and faster failure detection time on LAGs. BFDs’ lightweight and fast protocol allows for no traffic loss when used along other protocols like LACP. Overall, this standard takes full advantage of BFD failure detection and the high bandwidth capacity of LAGs. A standard like this is extremely useful and needed in today's networks as higher traffic loads are able to be handled and managed while ensuring reliability.

BFD on LAG Networks

LAGs are used in many networks today as they keep up with the high traffic and data rate demands. As mentioned by the authors of the research paper “Methods for localizing link failures," there are no standard protocols to localize link failure (LLF) at the physical layer [8]. In this paper we have analyzed BFD and LACP. These protocols work on a local level, this means that the detection and handling of failed links is done within the protocols and components connected by this link. The states or information of what member link failed is not communicated to higher layers as there exists no standard. This task is left for developers and researchers to design, study, and develop various methods.

In networks it is important to localize a link failure to keep network performance and reduce packet loss. By being able to localize a link failure packet a network can reroute packets, re-evaluate routing to ensure network capacity is used properly, and quickly fix the failed link. The aforementioned authors evaluate 6 methods for LLF. These methods are classified into two sets based on mechanisms: correlated-paths probing and per-link monitoring. The correlated-paths probing LLF mechanisms find link failure by sending a probe packet and correlating returned packet results at a centralized monitoring point/s. The authors mention that these mechanisms suffer from scalability problems, lack of multi-link failure support, and the lack of flexibility when network topology changes. These set of methods still are able to support, adapt, and localize failure but require extensive computation. The per-link monitoring LLF mechanisms find link failure by assigning one probe to test one specific link in the network. These mechanisms have two types of localization: proactive and reactive. Proactive localization uses a protocol like BFD to test the link periodically. In comparison reactive localization waits for the monitoring node to request to test the link. The advantages to per-link monitoring are that the probe-packet size is small using less network capacity and it supports multi-link failure detection. The authors mention mechanisms using probing might suffer from a high amount of probe packets, but this could be mitigated with multiple monitoring nodes spread over the network. The performance indicators used in this paper are scalability, detection time, localization accuracy, flexibility, and applicability in the network. One performance indicator that stands out is localization accuracy. The authors characterized two aspects of localization accuracy: the ability to localize link failure at the physical layer and the ability to localize multiple links that fail at the same time. It is interesting that the two per-link probing implementations that used BFD for LLF had the best accuracy and latency. Out of these two implementations one has the highest ratings in 3 of the 5 performance indicators. This demonstrates the core purpose of BFD in a LAG network as it is a lightweight protocol for fast failure detection.

The authors concluded that no one implementation, in the two classified groups, was the overall winner. This leaves room for further development, design, and testing. Even though the core purpose of BFD on LAG is being used, various implementations could introduce interoperability issues and unforeseen security vulnerabilities for vendors and users.

Challenges

RFC 7130 specifies that BFD should be able to detect link failures in a LAG irrespective of whether LACP is running. However, it does not specify the details of it. This might require that BFD and/or LACP engines know or communicate the state of each other. The exact mechanisms that enable BFD and LACP to share the details about their respective states with each other are not specified in any standards. This leads to differences in implementation of interoperability between BFD and LACP by multiple vendors. An example of this is the scenario when BFD over LAG feature is being provisioned after the link was already active in the LAG. In certain implementations, this is handled by retaining the link in its active status to avoid disruption to traffic forwarding while the micro BFD session is brought up. However, other implementations may not follow this method. Similarly, when BFD over LAG feature is unprovisioned after the links were already BFD validated and active within the LAG, certain implementations retain their links in their active status to avoid disruption to traffic forwarding. However, not all implementations follow this method. This results in interoperability issues.

Conclusion

LAG bundles multiple physical links between two adjacent nodes and allows this aggregation to appear as a single high bandwidth virtual channel to higher-layer protocols. Even though LACP includes provisions to detect failures in links, it is a complex protocol and not suitable for fast failure detection. On the other hand, BFD is a lightweight protocol that is capable of faster detecting failures in links of a LAG. While it is a strength of BFD that it works regardless of whether LACP is running in a LAG or not, that also introduces challenges due to the requirement of BFD and LACP machines to share the details about their states with each other. There are also challenges due to non-uniform handling of local BFD states when the remote BFD is provisioning or un-provisioning with its LAG. As there are multiple implementations to handle these scenarios, we hope that standardization committees will proactively create new standards to avoid interoperability issues, which will make the use of BFD on LAG more widespread in the networking world.

References

[1] J. Nielsen. Nielsen’s Law of Internet Bandwidth. Neilsen Norman Group. 2019.
[2] M. Bhatia. Issues with how BFD is currently implemented over LAGs. 2011.
[3] Fast ReRoute error detection - Implementation of BFD mechanism, Jozef Papan, *et al.*, IEEE 2019.
[4] RFC 5880 - Bidirectional Forwarding Detection (BFD), IETF 2010.
[5] IEEE 802.1AX - IEEE Standard for Local and metropolitan area networks - Link Aggregation.

[6] Link Aggregation according to IEEE 802.3ad, white paper by SkyConnect GmbH, 2002.

[7] RFC 7130 - Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces, IEFT 2014.

[8] A. Basuki, F. Kuipers, Delft. Methods for localizing network link failures. 2018.

[9] IEEE Std 802.3ad-2000 Amendment to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications— Aggregation of Multiple Link Segments.