Preliminary Implementation of "Bidirectional Forwarding Detection" Protocol for Unicast and Multicast

THE TECHNICAL UNIVERSITY of CLUJ-NAPOCA

COMODICI

A.F Gabor *, V. Dobrota *, K. Steenhaut ** and T. Blaga *
* Technical University of Cluj-Napoca, Cluj-Napoca, Romania
** Vrije Universiteit Brussel/ETRO Department, Brussels, Belgium

Introduction

In the existing and future intelligent buildings, the major network technique for delivering information will be multicasting. This will be used to send both user data and signaling, so the impact of protocols able to detect problems and to help in troubleshooting is becoming higher.

Bidirectional Forwarding Detection (BFD), under standardization process by IETF, as *draft-ietf-bfd-base-07.txt*, issued in January 2008 is able to detect problems in a bidirectional link between two routing engines including interfaces, data link and eventually the routing engine itself. It is designed to obtain lower latencies, independent to medium, transmission and routing protocols. It could perform path failure detection for:

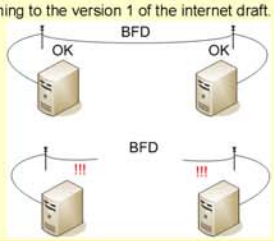
- physical links
- tunnels
- virtual circuits
- label switched paths in MPLS
- routed paths over several routers
- unidirectional paths

BFD will send signaling to its subscribers, monitoring the activation and interruption of any session. There are two operating modes:

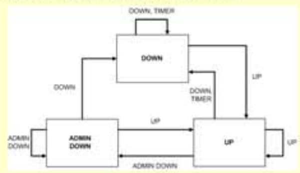
- primary (asynchronous) mode
- on demand (by request) mode

BFD for Multipoint Networks (mBFD)

The initial application was written in C for Linux by Tom Phelan from Sonus Networks Inc in October 2003. It has the following characteristics: it receives the connection parameters from the command line; it supports only one connection per machine; it does not implement the "echo" function; it has been written confirming to the version 1 of the internet draft.



The only modification brought to the current version of the header is the meaning of some bits.

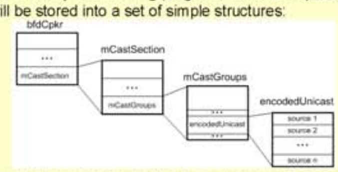


The new mBFD state machine has only three states compared to the BFD one, which has four states. Actually the State 2: Init was removed for simplicity reasons. Each system will communicate its state using a field within mBFD packets.

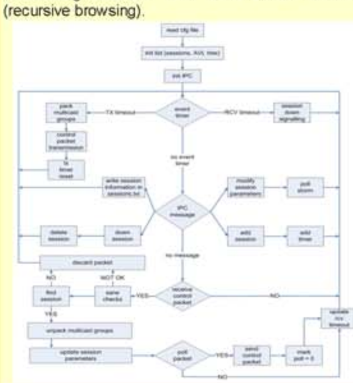
The application is structured on some components:

- session list
- timer tree
- communication descriptors (sockets) array
- pointer-to-functions array
- functions

The session list is a single-linked list of structures of type `bfdSession` that contains all the needed information about the state of a session at a given point in time. Using this structure we can consider each session as an object. The great majority of the written functions contain as a parameter a `bfdSession` structure. By passing as a parameter the session on which we want to operate, we can effectively simulate objects, their attributes and actions with the simplicity of the C language. Within the memory of a running program, the control packets will be stored into a set of simple structures:



The memory space needed to send and to receive the control packets are simple uint8_t buffers. At receiving side the functions invoked will return the number of bytes actually read, and so we can allocate the exact amount of memory needed. After this is being done, the fields which were described above will be read from the buffer one after the other, in the exact order in which the fields show up in the `bfdCpkt` structure, descending into the next level when necessary (recursive browsing).



Conclusion

Although it is a part of a complex project developed at Vrije Universiteit Brussel, the current version of mBFD proposed in this paper is a standalone program, but it could be integrated soon as a software module for multicast routing protocols such as PIM-SM, PIM-DM. It will fasten their convergence, by decreasing the detection time from few minutes down to 1-2 seconds. Performances could be even better because the transmission periods and detection multipliers make the detection time as low as few milliseconds. The evaluations are in a preliminary stage and the above mentioned parameters may need a better accuracy and more testbed scenarios. According to preliminary measurements, mBFD does not seem to be efficient in backbone networks. However it might be efficient in access networks for newly deployed applications for intelligent buildings, involving both unicast and multicast. A particular advantage could be obtained for video streaming (i.e. IPTV) if the control packets are sent separately to avoid a session failure simultaneously with data packets. Rapid detection of service discontinuity will allow fast re-routing.

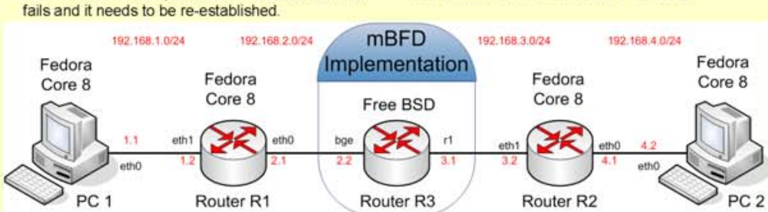
Experimental results

TEST 1

- 96-byte control packets from router R1 (49142/UDP) through router R3 to router R2 (3785/UDP).
- The transmission time XMT was decreased from 10 ms up to 1 μs (detection multiplier LDM = 5).
- During the experiment that lasts according to XMT if more than 5 packets are lost, the session fails and it needs to be re-established.

TEST 2

- included in addition two PCs, running FC8. 300-byte packets with an average service rate of 100,000 packets/second, from 5001/UDP to PC2, on the route: PC1-R1-R3-R2-PC2.
- network traffic generator/receiver, `mgen-4.0x3`, was launched at both PC1 and PC2.
- the failure occurred when XMT ≤ 2.6 ms.



XMT [ms]	DET [ms]	Test 1	Test 2
10.000	50.000	Stable	Stable
5.000	25.000	Stable	Stable
2.500	12.500	Stable	Fail & re-stable
1.250	6.250	Stable	Fail & re-stable
0.625	3.125	Stable	Fail & re-stable
0.400	2.000	Stable	Fail & re-stable
0.312	1.560	Fail & re-stable	Fail & re-stable
...	...	...	...
0.001	0.005	Fail & re-stable	Fail & re-stable

DET = LDM * RTX

LTX/RTX local/remote transmission intervals
LRX/RRX local/remote minimum receive intervals
LDM/RDM local/remote detection multipliers
XMT (calculated transmission time)
DET (calculated detection timer)