

libosmocre - Bug #4836

ns2: load sharing function for UDP data

10/27/2020 05:56 PM - lynxis

Status: New	Start date: 10/27/2020
Priority: High	Due date:
Assignee: lynxis	% Done: 0%
Category: libosmogb	
Target version:	
Spec Reference:	
Description	
Implement the load sharing function for UDP data.	
<ul style="list-style-type: none">on a new LSP select a NSVC based on data weight e.g. count all weight as sum_weight, LSP %modulo sum_weight, select the NSVC.	
NSVC1: data weight: 10 NSVC2: data weight: 5	
sum = 15, LSP%sum in (0- 9) => NSVC1 sum = 15, LSP%sum in (10-14) => NSVC2	
Maintain a LSP table per NSE containing "LSP, NSVC, timestamp/counter". The timestamp is used to remove old entries when the table is full.	

History

#1 - 10/27/2020 05:59 PM - daniel

- Description updated

#2 - 11/14/2020 08:42 AM - laforge

- Category set to libosmogb

#3 - 11/14/2020 08:53 AM - laforge

- Assignee set to lynxis

- Priority changed from Normal to High

Related: load sharing function for FR.

In the FR case, we should assume all weights are equal, and all NS-VC are equally suitable for signaling and user plane traffic.

In the TTCN3 code I chose to simply do a modulo operation of the identifier (LSP) into the array of currently unblocked NS-VCs. That of course means that every time links go up and down there will be a re-assignment of LSP to NS-VC. Some of that would be avoidable in your list-based approach, but I would argue we can get away with the modulo-based approach. After all, when NS-VCs go down, we inevitably will have a re-distribution of traffic among the remaining NS-VCs. Your approach reduces the number of users/flows that gets redistributed. Not sure if it's worth the effort. The biggest danger in changing NS-VC for an established connection is re-ordering of a few messages.