节点稳定性预测

Report - 3

高等理工学院
王冠楠
wgn2015@buaa.edu.cn

关于调试

- 在 aodv-uu 文件夹下 make clean + make, 再在 ns-2.35 下 make, 可以缩短编译所需时间。
- 修改 ns-2.35/ tcl/ lib/ ns-default.tcl

```
# AODV-UU routing agent
# unidir_hack_ = 0
Agent/AODVUU set unidir_hack_ 1
Agent/AODVUU set rreq_gratuitous_ 0
Agent/AODVUU set expanding_ring_search_ 1
Agent/AODVUU set local_repair_ 0
Agent/AODVUU set receive_n_hellos_ 0
Agent/AODVUU set hello_jittering_ 0
Agent/AODVUU set wait_on_reboot_ 0
Agent/AODVUU set debug_ 1
#                       0 here
Agent/AODVUU set rt_log_interval_ 0
Agent/AODVUU set log_to_file_ 0
Agent/AODVUU set optimized_hellos_ 0
#                                  0 here
Agent/AODVUU set ratelimit_ 1
Agent/AODVUU set llfeedback_ 0
#                              1 here
Agent/AODVUU set internet_gw_mode_ 0
```

```
/* Schedule the first HELLO */
if (!llfeedback && !optimized_hellos)
    hello_start();
```

参数计算的位置

- 在生成 hello 包之前计算之前时刻的参数，加入 hello 包中。

```
rrep = rrep_create(flags, 0, 0, DEV_NR(i).ipaddr,
                   this_host.seqno,
                   DEV_NR(i).ipaddr,
                   ALLOWED_HELLO_LOSS * HELLO_INTERVAL);
```

- 参数放在 hello 包的哪个位置上?

邻居统计

- 遍历路由表，找到所有距离为 1 跳且 valid 的节点，存入。

```
int NS_CLASS rt_calc_neighbor()
{
        int nc = 0; // counts of neighbor
        int i;
        for (i = 0; i < RT_TABLESIZE; i++) {
                list_t *pos;
                list_foreach(pos, &rt_tbl.tbl[i]) {
                        rt_table_t *rt = (rt_table_t *) pos;
                        // distance = 1, and should be valid
                        if (rt->hcnt == 1 && rt->state == VALID) {
                                fprintf(stderr,"     -> neighbor seqno: [%d]\n",rt->dest_seqno);
                                nc++;
                        }
                }
        }
        fprintf(stderr,"     -> number of neighbors: %d\n",nc);
        return nc;
}
```

邻居变化

- 每当一个节点的邻居变化时 / 发送 hello 消息时，计算该参数。
- 用两个全局的数组存储，每个节点存储上时刻和当前时刻的邻居。
- 用 ip 地址区分邻居节点。

```
/*   neighbor change   */
    fprintf(stderr,"debug-> begin to calc neighbor in aodv-hello\n");
    fprintf(stderr,"     -> current node:%s\n",DEV_NR(i).ipaddr);
    n_nn = rt_calc_neighbor();
    float change = l_nn == 0 ? 1 : ( 1 - n_nn / l_nn > 1 ? 1 : 1 - n_nn / l_nn);
    fprintf(stderr,"                    -> pst: %d\n",n_nn);
    fprintf(stderr,"                    -> old: %d\n",l_nn);
    fprintf(stderr,"                    -> chg: %lf\n",change);
    l_nn = n_nn;
```

```
INITIALIZE THE LIST xListHead
INITIALIZE THE LIST xListHead
INITIALIZE THE LIST xListHead
created nodes
0.0.0.0: 00:00:00.000 hello_start: Starting to send HELLOs!
-> *** hello send!
0.0.0.0: 00:00:00.000 startAODVUUAgent: Routing agent with IP = 0.0.0.0 : 0 started.
0.0.0.0: 00:00:00.000 startAODVUUAgent: Settings:
0.0.0.0: 00:00:00.000 startAODVUUAgent: unidir_hack ON
0.0.0.0: 00:00:00.000 startAODVUUAgent: rreq_gratuitous OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: expanding_ring_search ON
0.0.0.0: 00:00:00.000 startAODVUUAgent: local_repair OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: receive_n_hellos OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: hello_jittering OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: wait_on_reboot OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: optimized_hellos OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: ratelimit ON
0.0.0.0: 00:00:00.000 startAODVUUAgent: llfeedback OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: internet_gw_mode OFF
0.0.0.0: 00:00:00.000 startAODVUUAgent: ACTIVE_ROUTE_TIMEOUT=3000
0.0.0.0: 00:00:00.000 startAODVUUAgent: TTL_START=2
0.0.0.0: 00:00:00.000 startAODVUUAgent: DELETE_PERIOD=15000
0.0.0.1: 00:00:00.000 hello_start: Starting to send HELLOs!
-> *** hello send!
0.0.0.1: 00:00:00.000 startAODVUUAgent: Routing agent with IP = 0.0.0.1 : 1 started.
0.0.0.1: 00:00:00.000 startAODVUUAgent: Settings:
0.0.0.1: 00:00:00.000 startAODVUUAgent: unidir_hack ON
0.0.0.1: 00:00:00.000 startAODVUUAgent: rreq_gratuitous OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: expanding_ring_search ON
0.0.0.1: 00:00:00.000 startAODVUUAgent: local_repair OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: receive_n_hellos OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: hello_jittering OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: wait_on_reboot OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: optimized_hellos OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: ratelimit ON
0.0.0.1: 00:00:00.000 startAODVUUAgent: llfeedback OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: internet_gw_mode OFF
0.0.0.1: 00:00:00.000 startAODVUUAgent: ACTIVE_ROUTE_TIMEOUT=3000
0.0.0.1: 00:00:00.000 startAODVUUAgent: TTL_START=2
0.0.0.1: 00:00:00.000 startAODVUUAgent: DELETE_PERIOD=15000
0.0.0.2: 00:00:00.000 hello_start: Starting to send HELLOs!
-> *** hello send!
0.0.0.2: 00:00:00.000 startAODVUUAgent: Routing agent with IP = 0.0.0.2 : 2 started.
0.0.0.2: 00:00:00.000 startAODVUUAgent: Settings:
0.0.0.2: 00:00:00.000 startAODVUUAgent: unidir_hack ON
0.0.0.2: 00:00:00.000 startAODVUUAgent: rreq_gratuitous OFF
0.0.0.2: 00:00:00.000 startAODVUUAgent: expanding_ring_search ON
```