

Em_canid ematch benchmark

Introduction

In order to evaluate the cost of our filter implementations, we conducted several experiments to measure the time spent in `can_send()` function (in the kernel) for different filter configurations.

Measurements were performed by using function graph tracer in `ftrace` – the Linux kernel function tracer. This particular tracer timestamps entry and exit points of the traced functions and stores the duration of the function execution. `Ftrace` was configured in dynamic mode (as shown later).

This allows us to trace only `can_send()` function, not affecting the performance of any other (not traced) functions.

Ftrace configuration:

```
FTRDIR=/sys/kernel/debug/tracing
sysctl kernel.ftrace_enabled=1
echo "function_graph" > ${FTRDIR}/current_tracer
sleep 1
echo "can_send" > ${FTRDIR}/set_ftrace_filter
echo 1 > ${FTRDIR}/tracing_on
```

The benchmarks were performed on an embedded computer with MPC5200 – embedded PowerPC CPU (e300 core, G2 LE), 396 MHz, with 128 MiB of RAM running 3.4.2 Linux kernel with custom `.config` file.

CAN traffic was generated with the command:

```
cangen can0 -I $ID -L 8 -D i -g $GAP -n 10000
```

where

- `$ID` sets CAN frame ID, it was set to the fixed value to be classified either by the first instance of a filter or the last one
- `$GAP` sets the delay in milliseconds between each sent frame – value 0 and 1 were used in the two test performed

Results

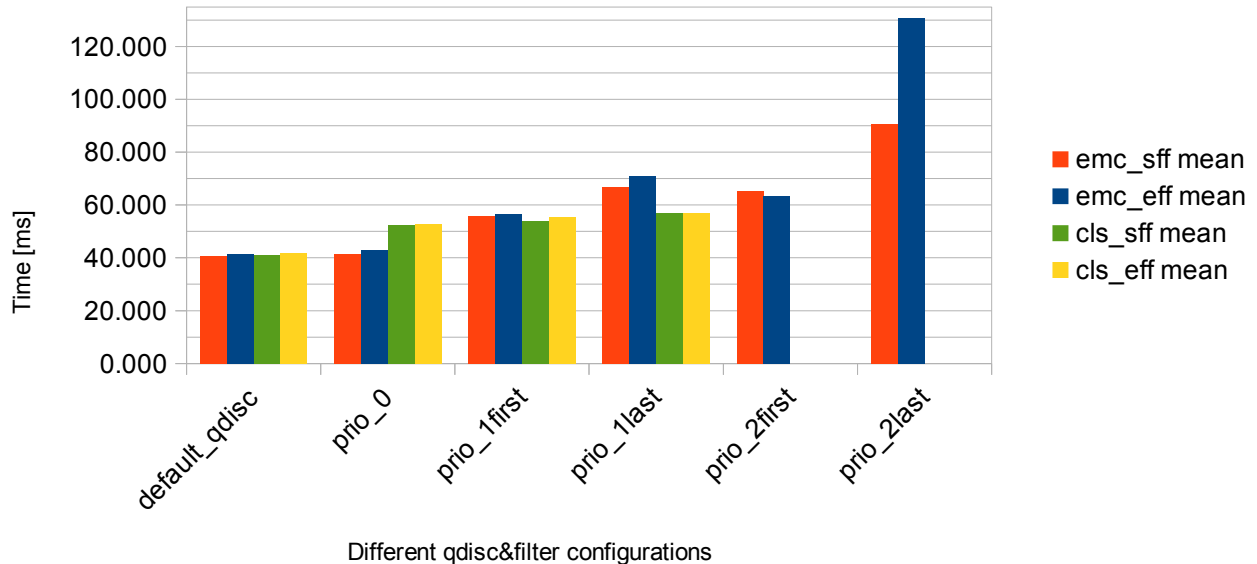
Measured values are in μs units. These values (*mean*, *min*, *max*) are counted from 10000 measurements of time spent in `can_send()` function.

	emc_sff			emc_eff		
	mean	min	max	mean	min	max
default_qdisc	40.541	39.392	130.944	41.502	40.160	163.392
prio_0	41.396	39.520	140.448	43.064	40.512	135.424
prio_1first	55.691	53.920	168.416	56.482	54.848	173.376
prio_1last	66.679	60.480	244.768	71.026	67.424	193.056
prio_2first	65.128	58.720	228.224	63.577	60.288	180.288
prio_2last	90.802	85.472	211.008	130.836	128.576	146.976

	cls_sff			cls_eff		
	mean	min	max	mean	min	max
default_qdisc	40.992	39.168	158.112	41.824	40.192	135.584
prio_0	52.564	50.848	169.760	52.865	50.976	178.112
prio_1first	54.041	52.672	226.528	55.422	53.728	172.896
prio_1last	57.071	54.720	176.000	56.952	54.816	169.376

Time spent in `can_send()` measured by ftrace

Can traffic generated with gap = 1 ms



Notation explanation:

- **emc_sff** – em_canid ematch used, SFF only frames generated by candump
- **emc_eff** – em_canid ematch used, EFF only frames generated by candump
- **cls_sff** – cls_can classifier¹ used, SFF only frames generated by candump
- **cls_eff** – cls_can classifier used, EFF only frames generated by candump

- **default_qdisc** – Default qdisc – nowadays quite always pfifo_fast
- **prio_0** – Prio qdisc with minimal configuration (2 bands – one for classification, one set as default)
- **prio_1first** – Prio qdisc with 10 bands – for each class (i.e. band) there is one classifier attached to it. All traffic is classified into the first class.
- **prio_1last** – Same as above, except all traffic is classified into the last class.

For em_canid only:

- **prio_2first** – Prio qdisc with 10 bands – each class has 1 classifier (basic) with 10 ematch rules (joined with OR) attached to it. All traffic is classified into the first class.
- **prio_2last** – Same as above, except all traffic is classified into the last class.

¹ cls_can – Stand-alone classifier of AF_CAN packets according to their identifiers. The purpose of this classifier was the same as of the em_canid, however the source code was twice as long with no known benefits.