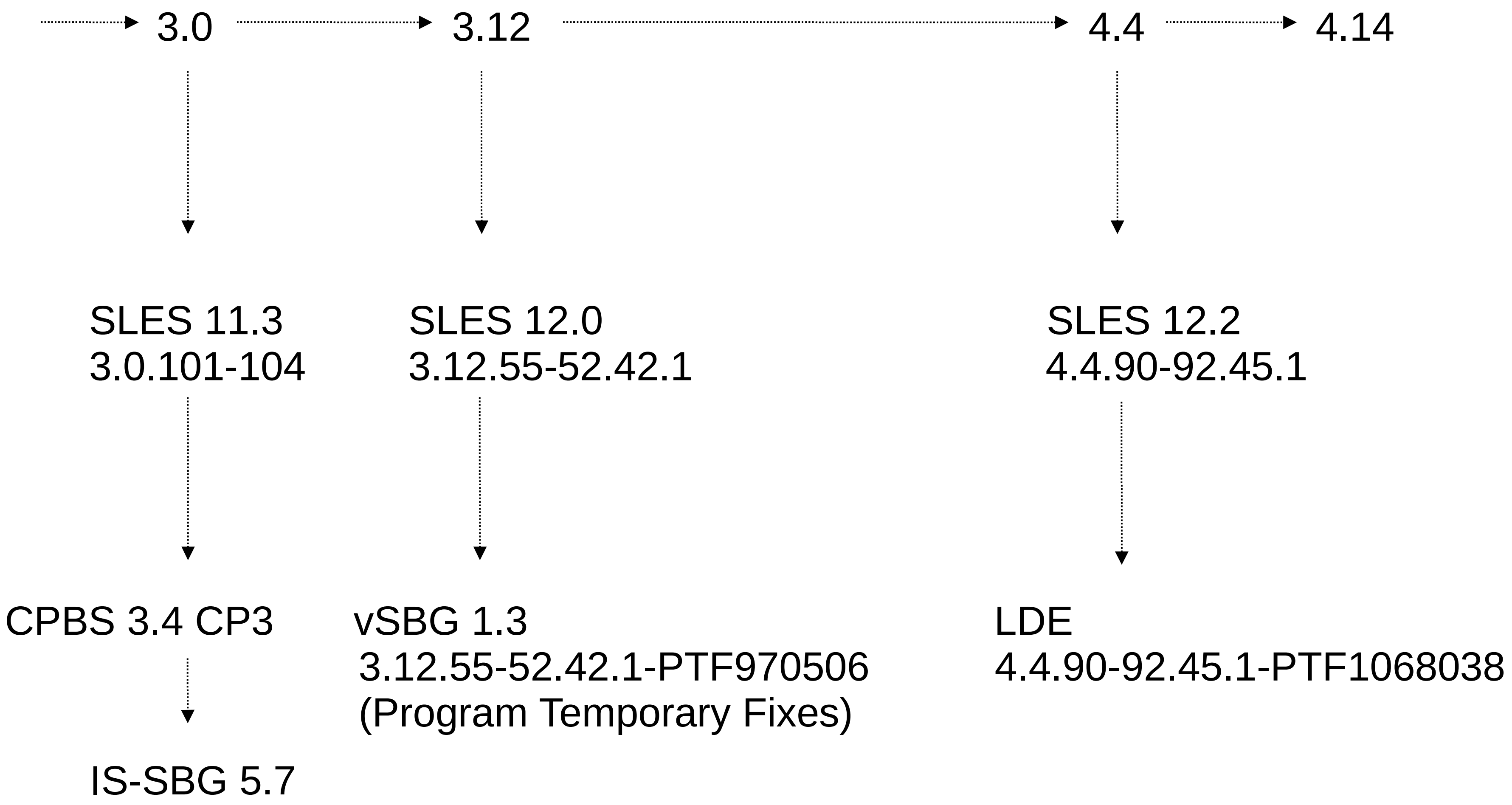
Kernel trouble shooting in SBG

1. Linux kernel intro
2. 5 SBG case studies
3. Trouble-shooting strategy
4. Kernel and LPO practicalities

Linux kernel

- The core of a general purpose operating system
- Created by Linus Torvalds 1991 on a 486 PC
- Ported to a multitude of platforms
- Inspired by various other UNIX systems
- Implements most of POSIX standards

Branching off the upstream kernel



3.0

3.12

4.4

4.14

SLES 11.3
3.0.101-104

SLES 12.0
3.12.55-52.42.1

SLES 12.2
4.4.90-92.45.1

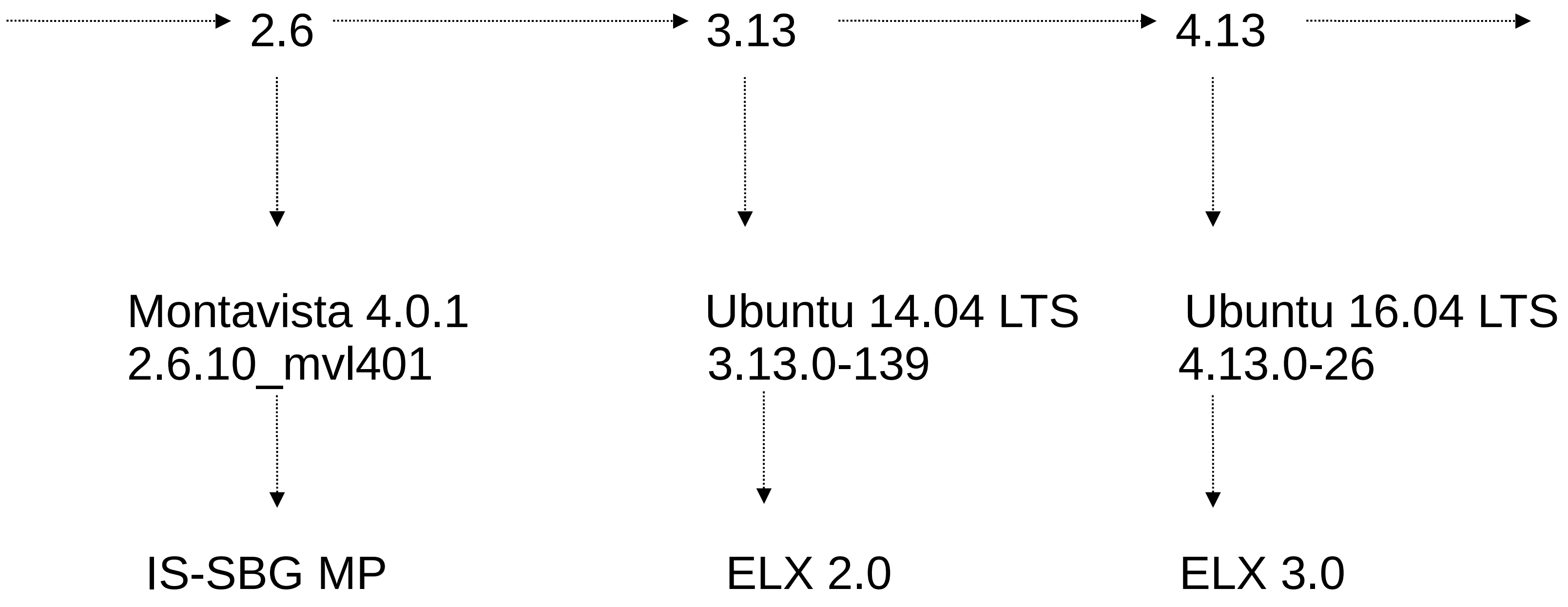
CPBS 3.4 CP3

IS-SBG 5.7

vSBG 1.3
3.12.55-52.42.1-PTF970506
(Program Temporary Fixes)

LDE
4.4.90-92.45.1-PTF1068038

Other distributions



2.6

3.13

4.13

Montavista 4.0.1
2.6.10_mvl401

Ubuntu 14.04 LTS
3.13.0-139

Ubuntu 16.04 LTS
4.13.0-26

IS-SBG MP

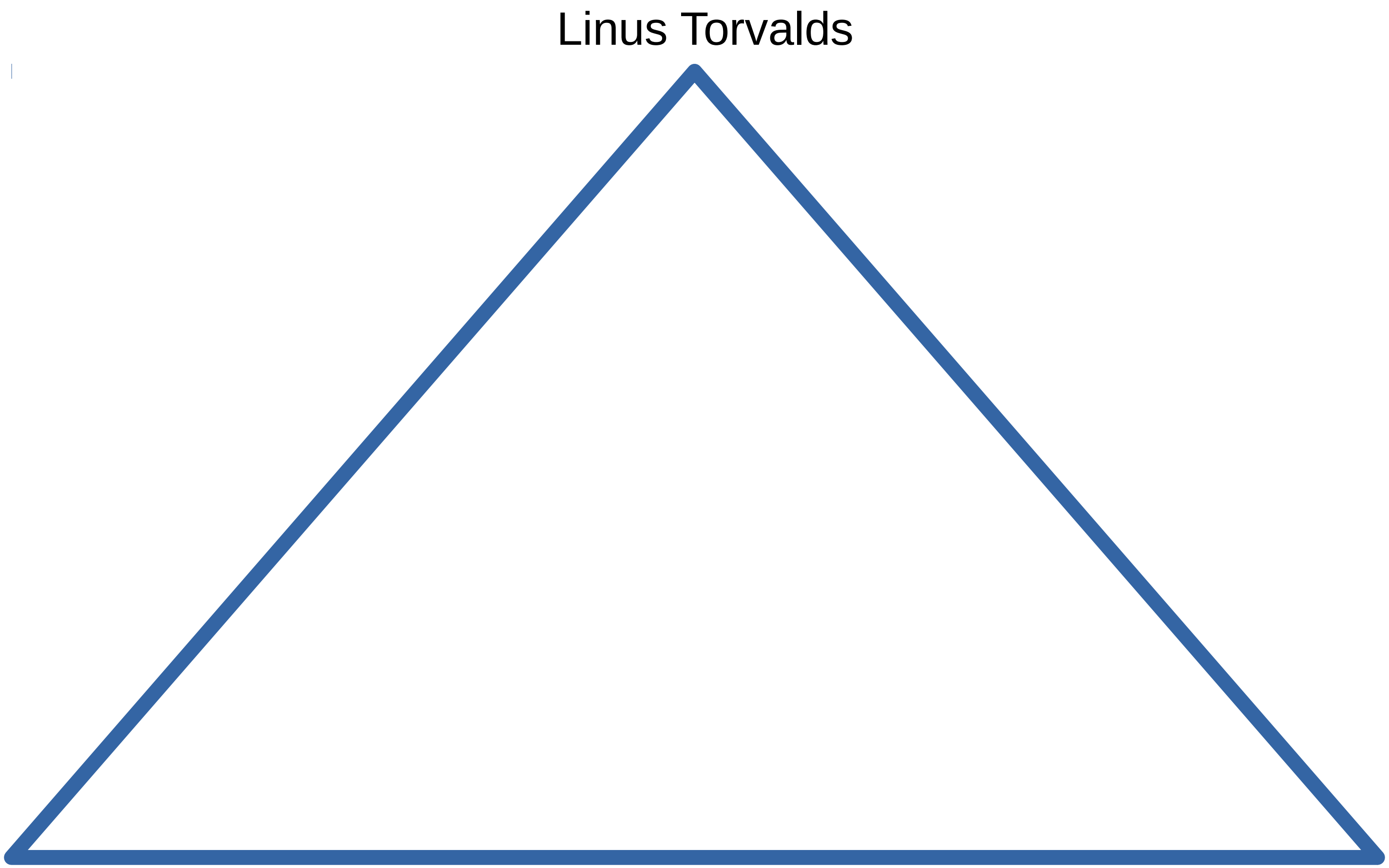
ELX 2.0

ELX 3.0

Kernel repos

- Kernel.org
  - git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
  - Tags v3.0, v3.12, v.4.4
- SUSE
  - git clone https://github.com/openSUSE/kernel
  - Tags v3.0, v3.12, v.4.4
  - Tags v3.0.101
  - Tags rpm-3.0.101-104 (the base for CPBS 3.4 CP3)
- Ubuntu
  - git clone git://kernel.ubuntu.com/virgin/linux.git

Linux kernel development hierarchy



FOSS

- Linux kernel have GNU license (GPLv2)
- Modifying kernel code in a commercial product can cause legal problems
- Ericsson general policy is to not modify kernel
- The purpose is to protect some patents

More FOSS

- Third-parties can implement patches for us
  - Montavista is the designated supplier
  - Only high-level functional descriptions can be given to Montavista (not code)
  - But we can point to existing patches on the Internet
- SUSE can provide corrections

FOSS exceptions

- Exception for FOSS can be granted
  - CPBS for IS-specific functionality (marker, ilfp)
  - EVIP in IS
- Modifying kernel in lab is ok

Support

- Jira LDC (General Linux/SLES support)
- Jira CC (LDE, escalations)
- CPBS (linux-wizards@dektech.com.au)
- SUSE (Corrections)
- Montavista
  - Magnus Nemell <mnemell@mvista.com>
  - Sharath Kurudi <skurudi@mvista.com>
- Jira GEP, Jira IS (SIS, MXB, ISER)

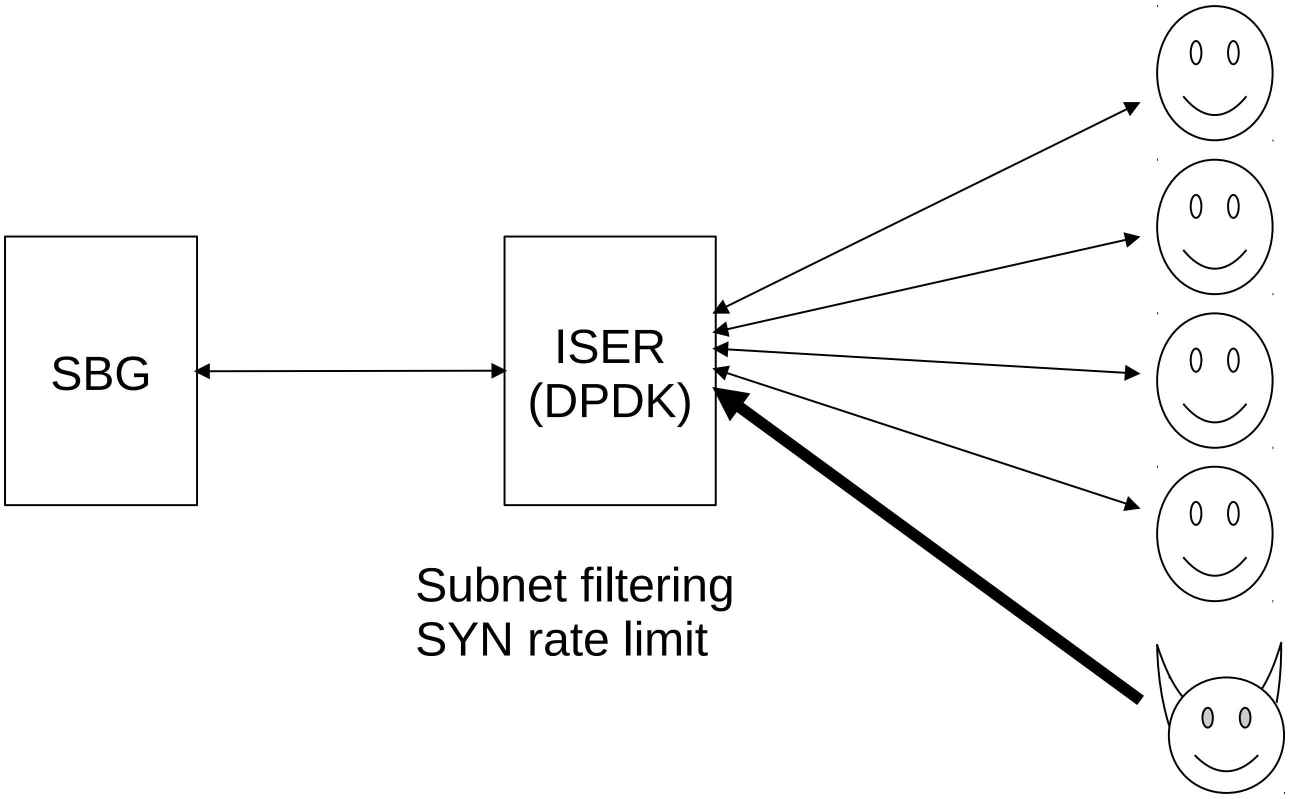
LDC/wizards

- Can be very helpful when given a good problem description
- But there are some things they cannot do
  - Reproduction
  - Building
  - Tracing
  - Interpret high-level problem observations

Case 1: IP route cache (rtcache)

- IPv4 – HV32435 – Telefonica, DT
- IPv6 – HU43605 – TMO

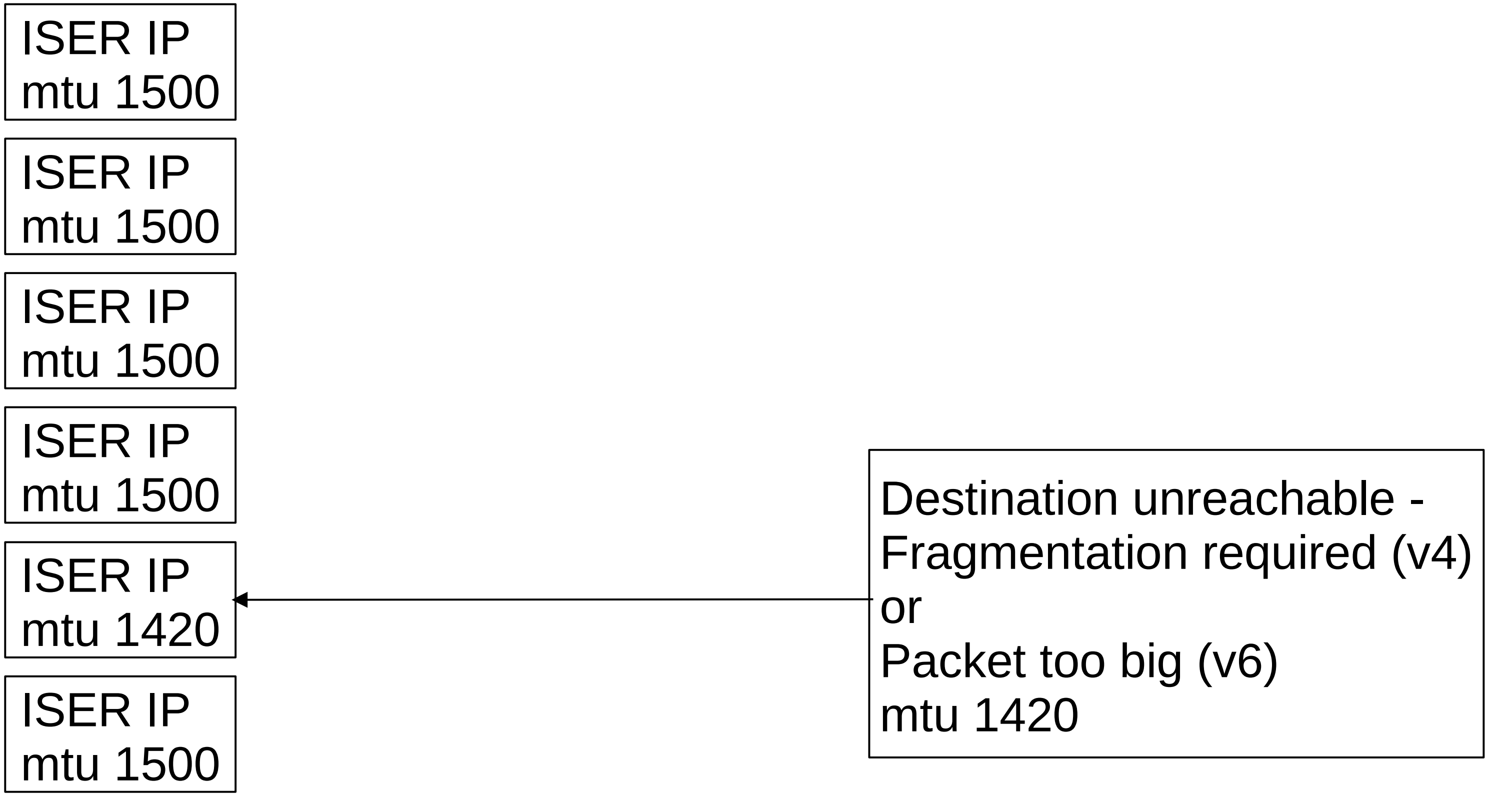
One route cache entry per source address (rtcache)



Subnet filtering
SYN rate limit

0-255.0-255.0-255.0-255

Duplicated information in route cache (rtcache)



ISER IP
mtu 1500

ISER IP
mtu 1500

ISER IP
mtu 1500

ISER IP
mtu 1500

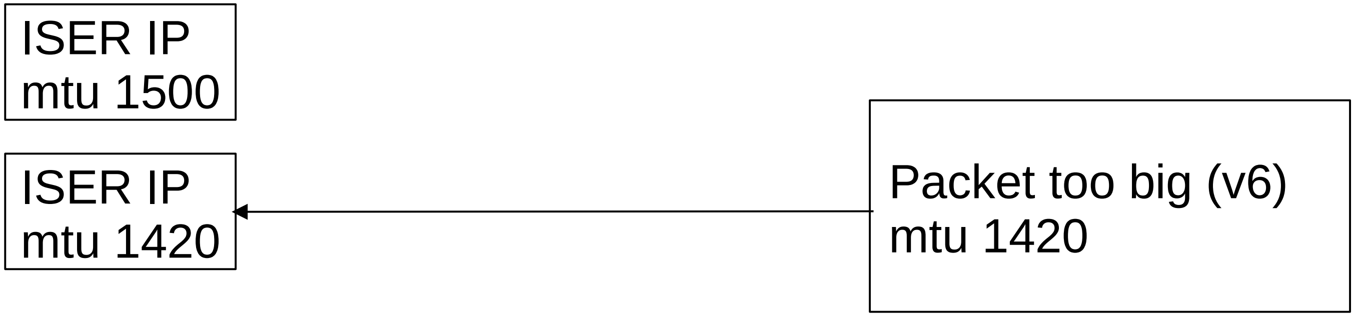
ISER IP
mtu 1420

ISER IP
mtu 1500

Destination unreachable -
Fragmentation required (v4)
or
Packet too big (v6)
mtu 1420

Upstream route cache changes (rtcache)

- v4 route cache removed
  - v3.5 (SLES 12 SP0)
- v6 route cache on demand
  - v4.1 (SLES 12 SP2) by facebook



IPv4 route cache (rtcache)

- SYN DoS attacks cause IP stack to become unresponsive and eventually some supervision mechanism will restart system
  - IS-SBG
    - Marker (RLSP) (default 3 x 10ms plus 3s)
    - Erlang net tick (8s)
    - Watchdog (16s)
  - vSBG
    - Erlang net tick (8s)
    - Watchdog (10s)

SYN DOS patch (rtcache)

- Make ingress SYN and egress SYN-ACK entries more eligible for removal
- When cache is over 50% full do not add new entries (thus avoiding excessive gc)

SYN DOS patch history (rtcache)

- 12B (R17A) Original patches introduced (max_size/2)
- 14B (R23B/1) Montavista's first attempt by our description (max_size/16)
- 5.1 (R29A) No SYN DOS patches (removed due to ACK issues)
- 5.2 (R30A) Montavista's second attempt (max_size/16)
- 5.6 (R34A) Original patches back again (max_size/2)
- 5.7 (R35A) Original patches now signed by Montavista (max_size/2)
- 5.8 (R36A) Original patches slightly modified (0.99 x max_size)
- 5.10 (R38A) The same signed by Montavista (0.99 x max_size)

IPv6 route cache (rtcache)

- Garbage collection every 30s can cause rejects
- Global gc lock cause all other cpu to wait

net.ipv6.route.gc_interval (rtcache)

- Default is 30s
- VR 0 – 79s (sysctl.conf.*)
- VR 1, 7, 13... – 71s
- VR 2, 8, 14... – 73s
- VR 3, 9, 15... – 47s
- VR 4, 10, 16... – 37s
- VR 5, 11, 17... – 53s
- VR 6, 12, 18... – 59s

sysVR.erl

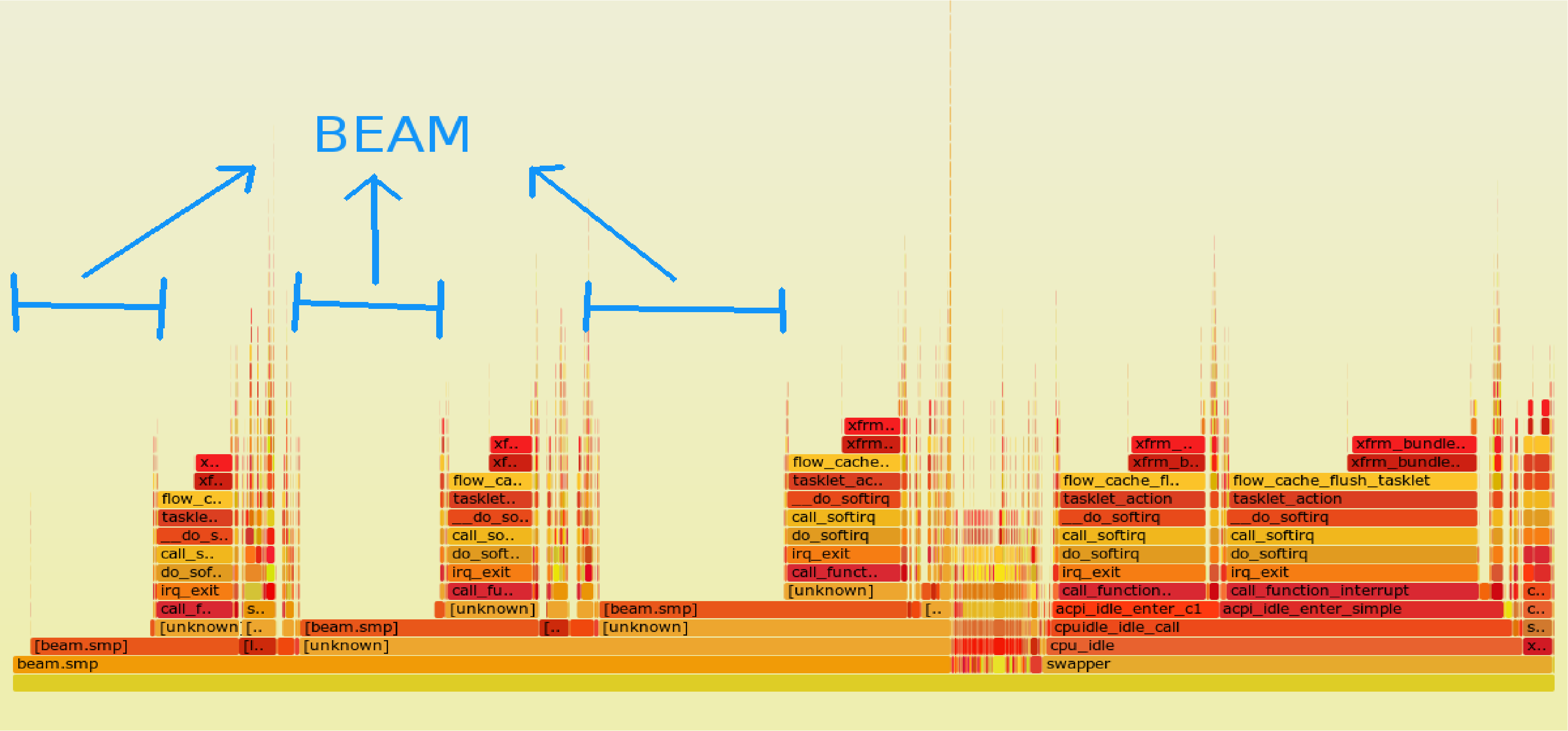
- FEE – 41s
- SE – 31s
- LBE – 43s

evip_startup_cmds.sh

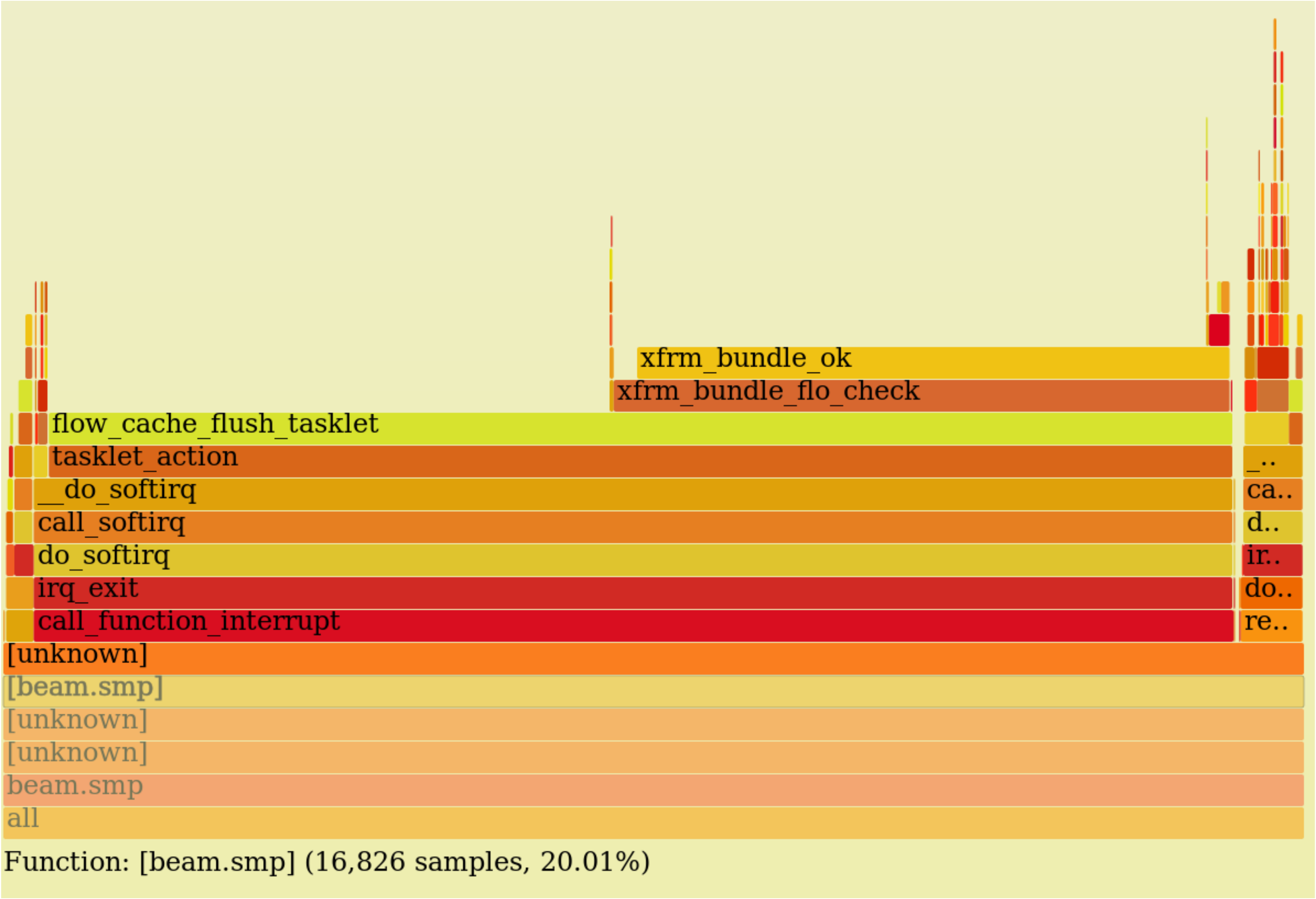
Case 2: UTRAN

- IS-SBG 5.5, HV10621
- Suddenly call rejects after a few weeks of traffic processing
- CPU spikes observed on rejecting call handler
- Once spikes start to appear they are quite frequent until next reboot

Perf (utran)


BEAM
x..
xf..
flow_c..
taskle..
__do_s..
call_s..
do_sof..
irq_exit
call_f..
s..
[unknown]
[..
[beam.smp]
[l..
beam.smp
xf..
xf..
flow_ca..
tasklet..
__do_so..
call_so..
do_soft..
irq_exit
call_fu..
[unknown]
[beam.smp]
[..
[unknown]
[beam.smp]
[unknown]
xfrm..
xfrm..
flow_cache..
tasklet_ac..
__do_softirq
call_softirq
do_softirq
irq_exit
call_funct..
[unknown]
[..
xfrm_..
xfrm_b..
flow_cache_fl..
tasklet_action
__do_softirq
call_softirq
do_softirq
irq_exit
call_function..
acpi_idle_enter_c1
xfrm_bundle..
xfrm_bundle..
flow_cache_flush_tasklet
tasklet_action
__do_softirq
call_softirq
do_softirq
irq_exit
call_function_interrupt
acpi_idle_enter_simple
c..
c..
s..
x..
cpuidle_idle_call
cpu_idle
swapper

Zoom in perf (utran)



Find culprit (utran)

- cd is-sbg-lpo/src/syf/lpo
- make
- cd out/tmp/kernel_build_dir/linux
- grep -rn xfrm_bundle_flo_check *

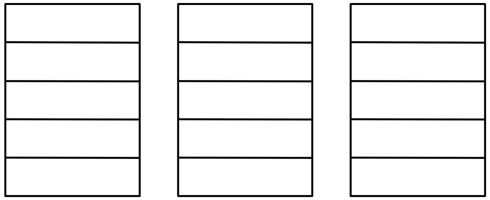
Why xfrm gc? (utran)

- Xfrm dst cache is nexthop information for each ipsec connection (IMS-AKA)
- Corresponding to ip route cache for non-ipsec
- Threshold for gc is sysctl net.ipv4.xfrm4_gc_thresh=2048000 and net.ipv6.xfrm6_gc_thresh=2048000 for GEP5
- SBG is well below even at max number of registered users
- SBG removes entries explicitly at unregister – no need for gc
- Number of entries is xfrm_dst_cache in slabtop

SLAB (utran)

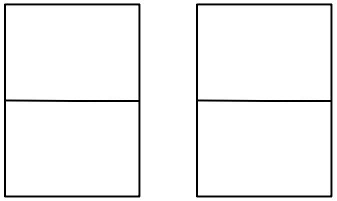
slabtop -o -sc

```
  OBJS ACTIVE  USE OBJ SIZE  SLABS OBJ/SLAB CACHE SIZE NAME
    24     24 100% 2048.00K     24        1     49152K kmalloc-2097152
 12102  12096  99%    0.64K   2017        6      8068K shmem_inode_cache
 23980  23617  98%    0.19K   1199       20      4796K dentry
  8127   7680  94%    0.55K   1161        7      4644K radix_tree_node
  4046   4046 100%    0.55K    578        7      2312K inode_cache
  [...]
```



cat /proc/slabinfo

```
# name            <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> ...
ip_vs_conn                0           0        384         10          1 ...
xt_hashlimit              0           0        104         37          1 ...
nfs_commit_data          11          11        704         11          2 ...
nfs_write_data          108         108        960          4          1 ...
[...]
```



grep xfrm_dst_cache /proc/slabinfo

```
xfrm_dst_cache            0      0     512    8    1 ...
```

What trigger xfrm gc? (utran)

Xfrm gc is called from one single place:

```
dst.c:dst_alloc(...)

if (ops->gc && dst_entries_get_fast(ops) > ops->gc_thresh) {

    if (ops->gc(ops))

        return NULL;

}
```

Search terms collected (utran)

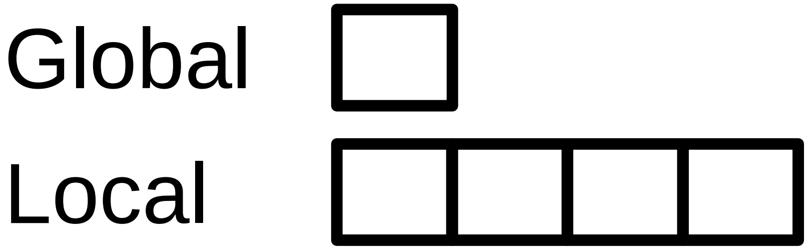
- percpu
- namespace
- xfrm

Commit a8a572a (utran)

```
commit a8a572a6b5f2a79280d6e302cb3c1cb1fbaeb3e8
Author:     Dan Streetman <dan.streetman@canonical.com>
AuthorDate: Thu Oct 29 14:51:16 2015
Commit:     Steffen Klassert <steffen.klassert@secunet.com>
CommitDate: Tue Nov 3 08:42:57 2015

    xfrm: dst_entries_init() per-net dst_ops
    […]
    The result of this is a very subtle bug; changes to the dst entries
    counter from one net namespace may sometimes get applied to a different
    net namespace dst entries counter
    [...]
```

Per-cpu counters (utran)

struct percpu_counter



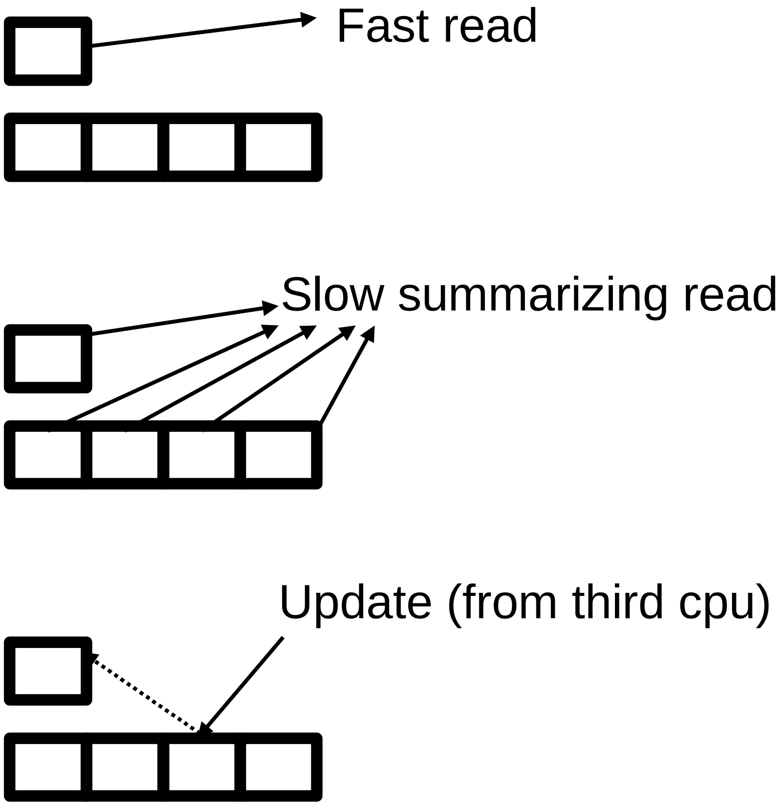
Global

Local

Batchsize 32

```
percpu_counter.h:
struct percpu_counter {
        raw_spinlock_t lock;
        s64 count;
#ifdef CONFIG_HOTPLUG_CPU
        struct list_head list;
#endif
        s32 __percpu *counters;
};
```

Per-cpu operations (utran)



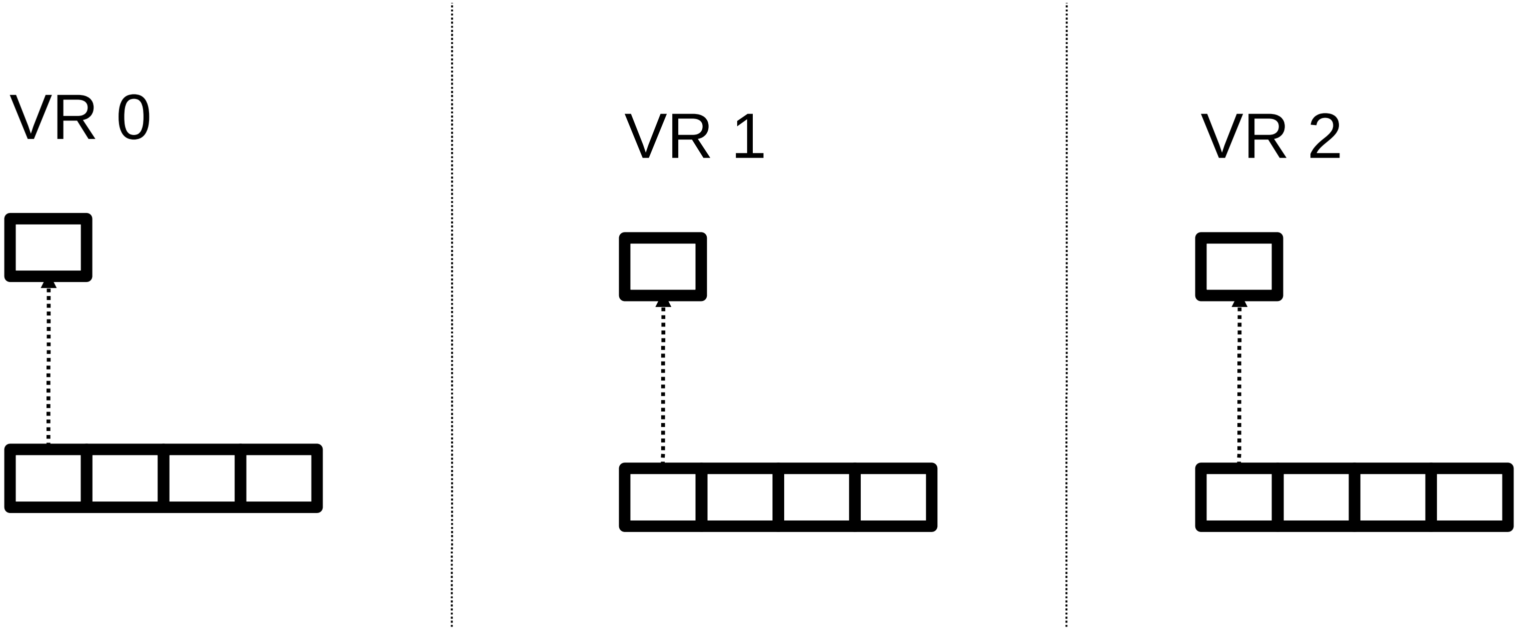
Fast read

Slow summarizing read

Update (from third cpu)

1. Update local
2. If local exceed batchsize then lock, add local to global and zero local

Multiple VRs (utran)

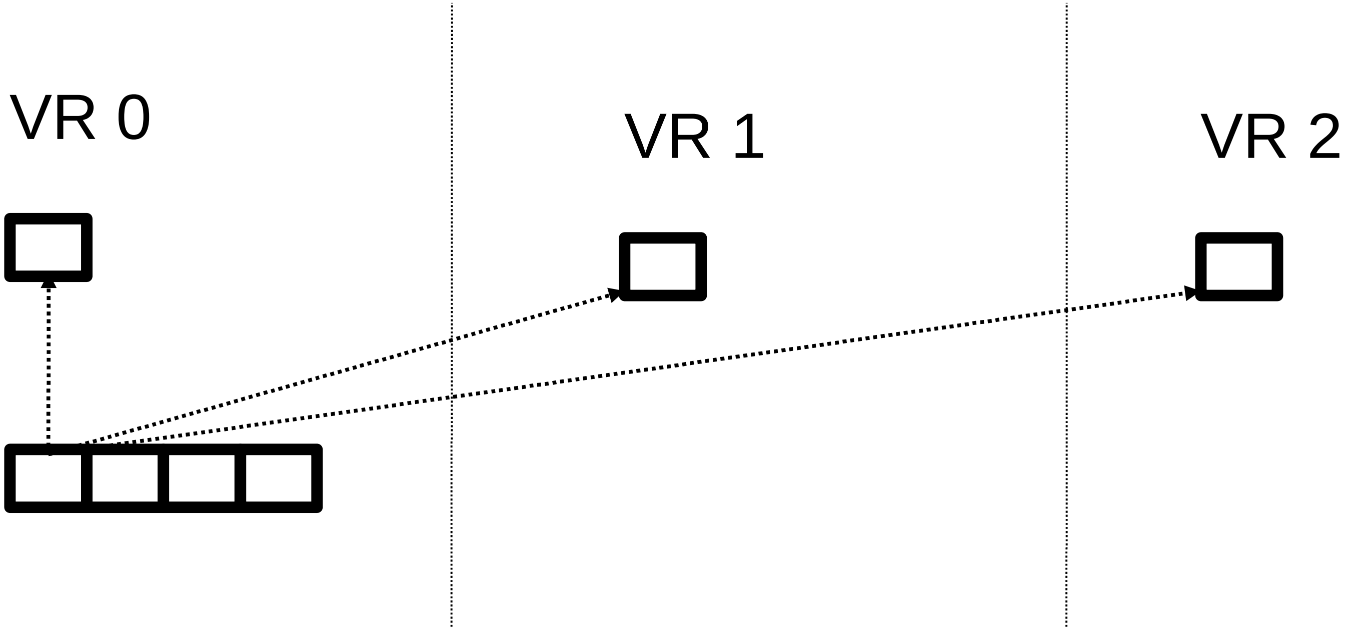


VR 0

VR 1

VR 2

Copy of pointer (utran)



Commit a8a572a (utran)

Old:

```
void __init xfrm4_init(void)
{
    dst_entries_init(&xfrm4_dst_ops);
    [...]
}

static void __net_init xfrm_dst_ops_init(struct net *net)
{
    struct xfrm_policy_afinfo *afinfo;

    rcu_read_lock();
    afinfo = rcu_dereference(xfrm_policy_afinfo[AF_INET]);
    if (afinfo)
        net->xfrm.xfrm4_dst_ops = *afinfo->dst_ops;
    [...]
    rcu_read_unlock();
}
```

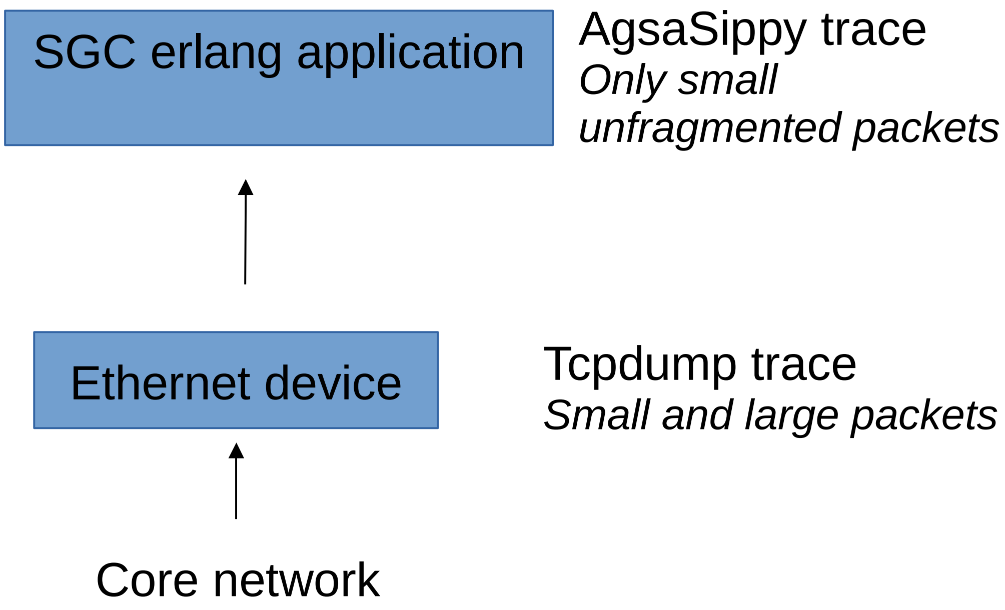
New:

```
static int __net_init xfrm4_net_init(struct net *net)
{
       int ret;

       memcpy(&net->xfrm.xfrm4_dst_ops,
              &xfrm4_dst_ops_template,
              sizeof(xfrm4_dst_ops_template));
       dst_entries_init(&net->xfrm.xfrm4_dst_ops);
       [...]
}
```

Case 3: UDP Fragment Reassembly

- vSBG 1.5 AT&T
- HW45917, HW38115, LDC-1119
- After upgrade 1.1 → 1.5 some INVITE from core not answered

Narrowing to fragments (reasm)



SGC erlang application

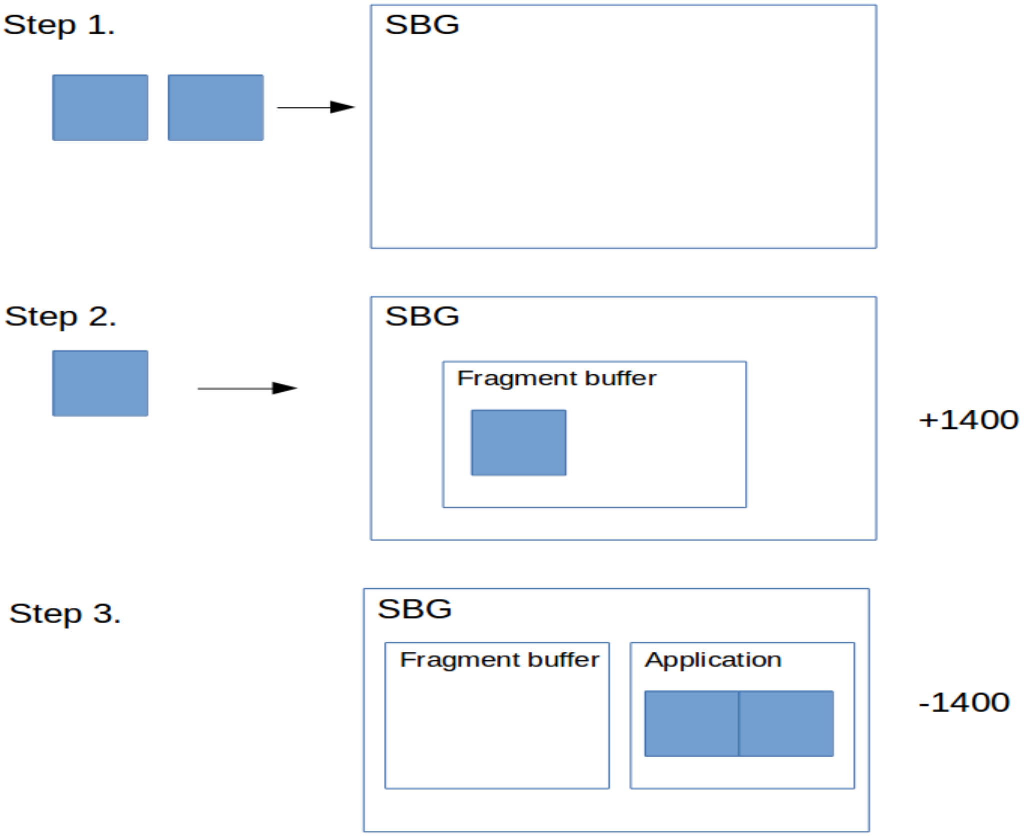
AgsaSippy trace
Only small
unfragmented packets

Ethernet device

Tcpdump trace
Small and large packets

Core network

Fragment buffer (reasm)



Fragment buffer size (reasm)

- If fragment buffer usage is above net.netfilter.nf_conntrack_frag6_high_thresh delete until below net.netfilter.nf_conntrack_frag6_low_thresh
- SLES 11.3 (vSBG 1.1)
  - Default thresholds high 256KB, low 192KB
  - Atomic integers
- SLES 12.0 (vSBG 1.5)
  - Default thresholds high 4MB, low 3MB
  - Percpu counters with batch size 130000
- allowFragments=false → high 0, low 0
- allowFragments=true → high 256KB, low 192KB

Lab Reproduction (reasm)

- Titansim bombarding with fragments at 300 cps
- With high threshold 256KB typically reproduced after 30-60min
- Later reproduced with 4MB after 7h

Kernel tracing (reasm)

- Establish that reassembly fails
- net/ipv6/reassembly.c is not used
- When ip6tables is used reassembly is done in net/ipv6/netfilter/nf_conntrack_reasm.c
- inet_frag_evictor() returns number of dropped fragments

Missing counters (reasm)

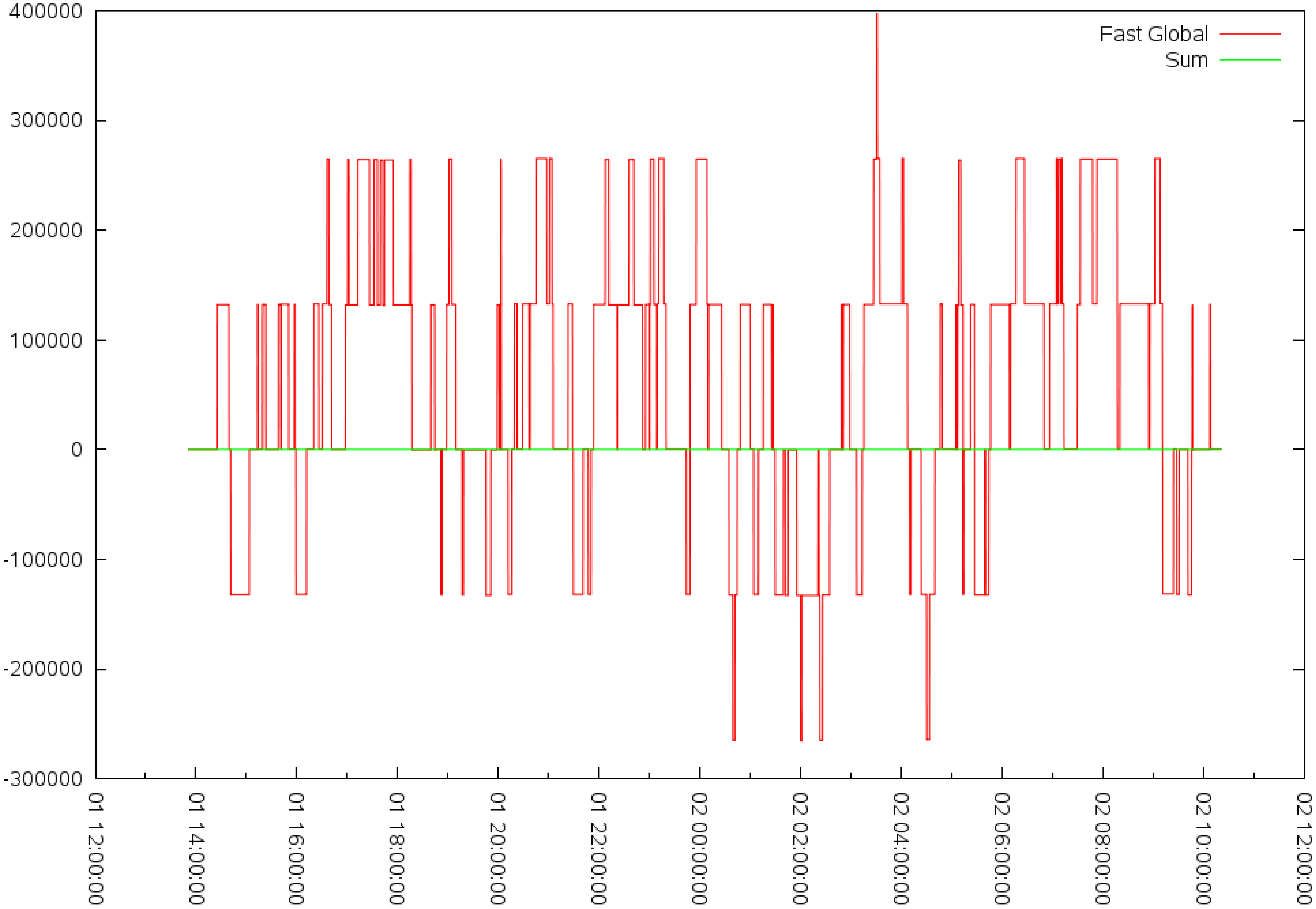
ss -s

```
Transport Total     IP        IPv6
*         0         -         -
RAW       0         0         0
UDP       22        14        8
TCP       115       96        19
INET      137       110       27
FRAG      0         0         0
```

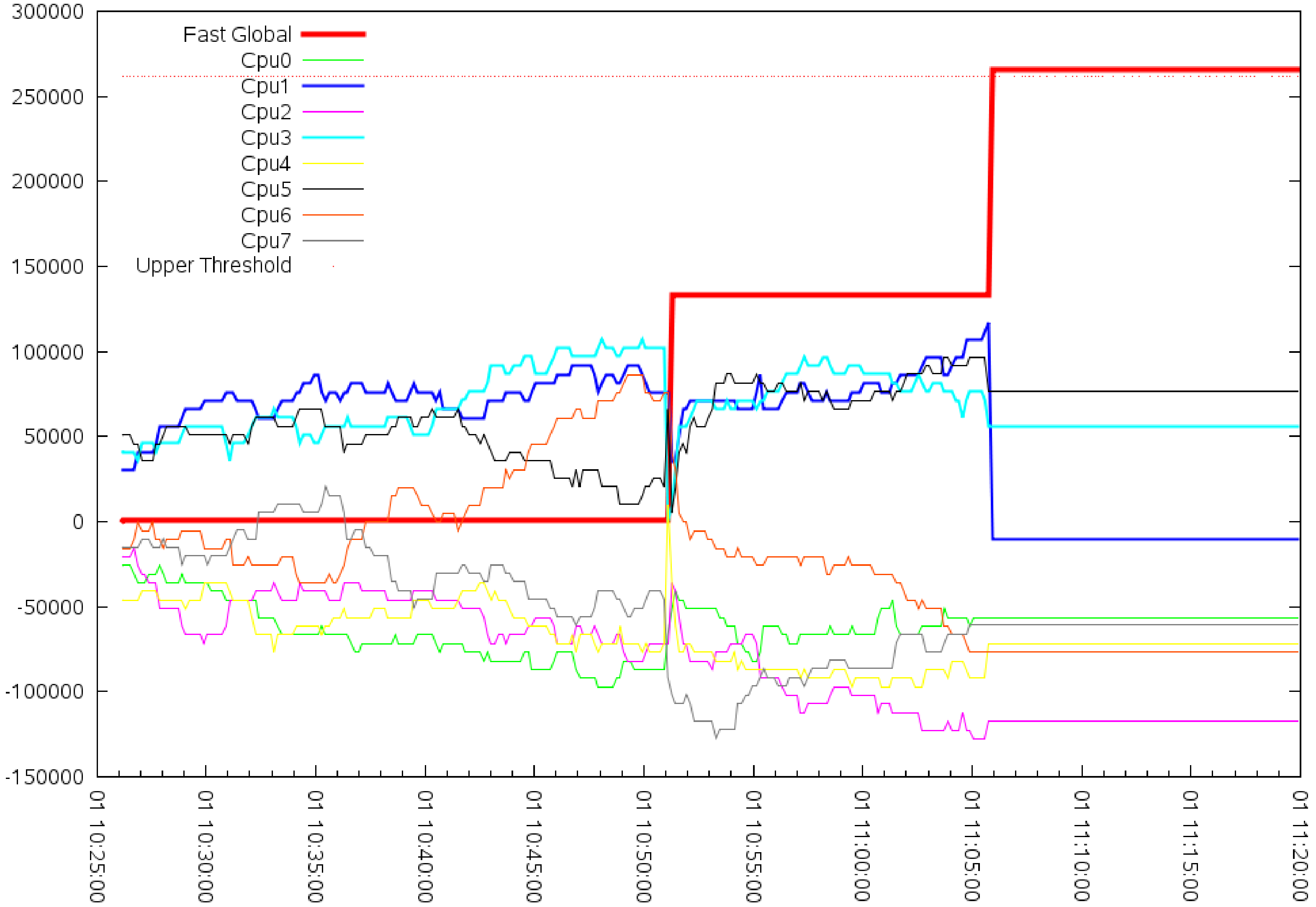
Ip6ReasmFails etc in netstat -s -A inet6

and /proc/net/snmp6

The fast value jumps up and down (reasm)



Once above thresh it is stuck (reasm)

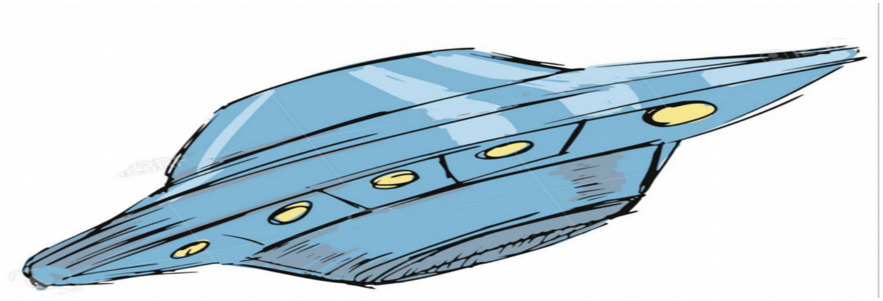


Correction (reasm)

- Increase allowFragments thresholds 4MB / 3MB
  - sysFirewall.erl
- Replace percpu counters with atomic integers
  - Also done in later kernels such as SLES 12.2
  - Patch received from SUSE

Case 4: Invalid fragments sent

- vSBG 1.5.1 AT&T
- HW53959, LDC-1147
- vSBG sends invalid UDP fragments



Fragment id (ufo)

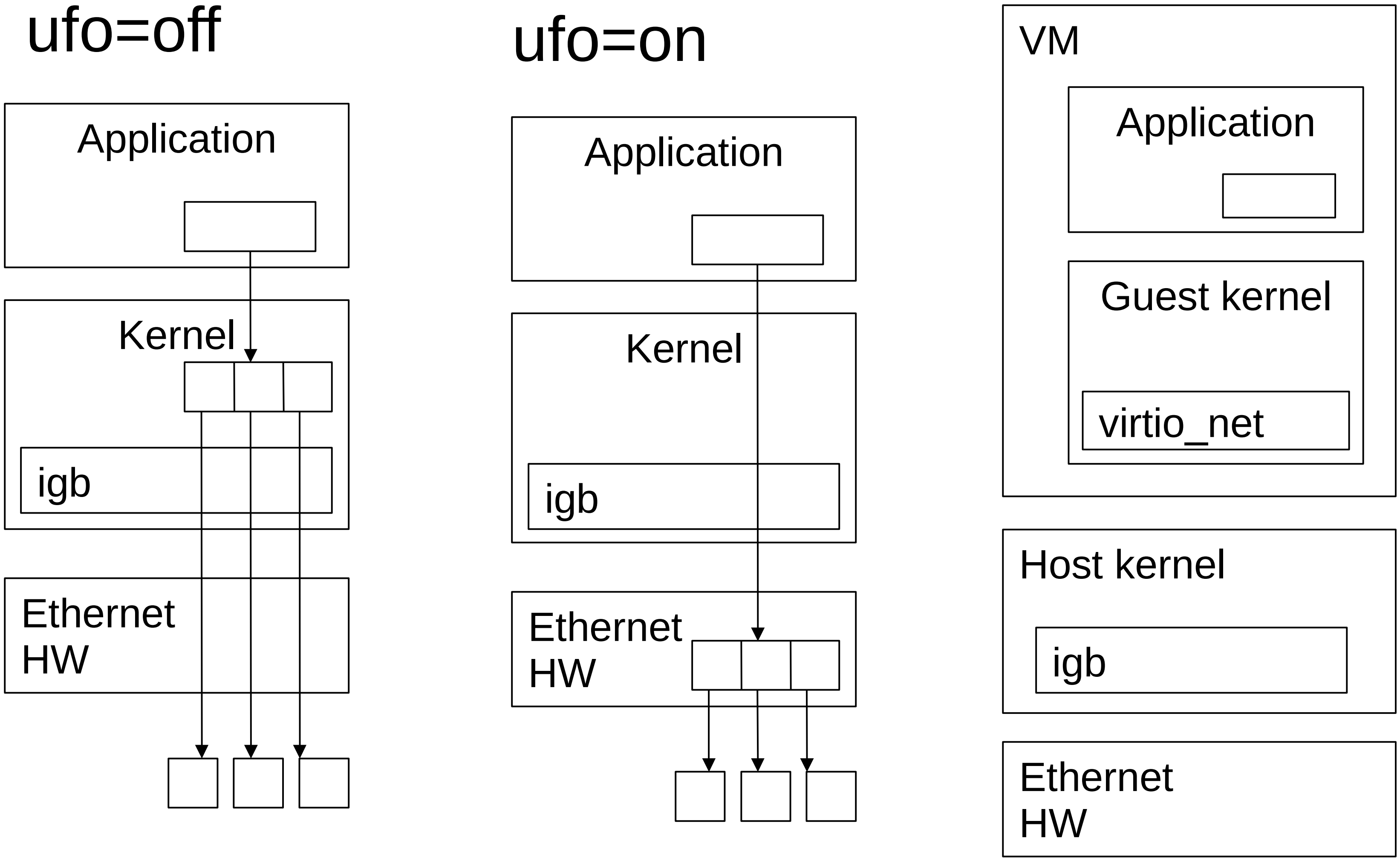
- Fragment id is zero in invalid packets (IP header)
- Upstream kernel have corrections for zero fragment id when udp-fragmentation-offload (ufo) is enabled
- Later upstream kernel have removed ufo completely for virtio_net driver
- AT&Ts cloud environment AIC have ufo enabled
- Ericssons CEE have ufo disabled

ethtool (ufo)

- ethtool -k eth1
- ethtool -K eth1 ufo off

```
Features for eth1:
rx-checksumming: off [fixed]
tx-checksumming: off
        tx-checksum-ipv4: off [fixed]
        tx-checksum-ip-generic: off [fixed]
        tx-checksum-ipv6: off [fixed]
        tx-checksum-fcoe-crc: off [fixed]
        tx-checksum-sctp: off [fixed]
scatter-gather: off
        tx-scatter-gather: off [fixed]
        tx-scatter-gather-fraglist: off [fixed]
tcp-segmentation-offload: off
        tx-tcp-segmentation: off [fixed]
        tx-tcp-ecn-segmentation: off [fixed]
        tx-tcp6-segmentation: off [fixed]
udp-fragmentation-offload: off [fixed]
generic-segmentation-offload: off [requested on]
generic-receive-offload: on
[...]
```

UDP Fragmentation Offload (ufo)



ufo=off

Application

Kernel

igb

Ethernet
HW

ufo=on

Application

Kernel

igb

Ethernet
HW

VM

Application

Guest kernel

virtio_net

Host kernel

igb

Ethernet
HW

virtio_net.c (ufo)

```
static bool gso = true;
module_param(gso, bool, 0444);
[…]
if (gso) dev->features |=
   dev->hw_features & (NETIF_F_ALL_TSO|NETIF_F_UFO);
```

Kernel parameters (ufo)

- vSBG repo lpo/lpo_addons/etc/config
- SC /home/tomte/sis/private/bs_SSC_1/tftpboot/pxelinux.conf

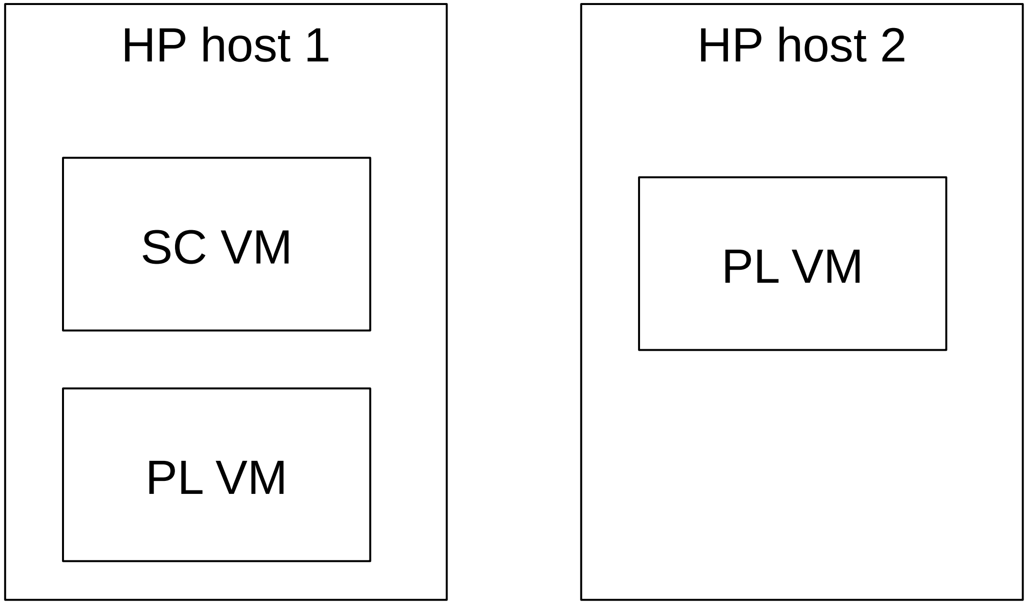
```
default vmlinuz

label vmlinuz

kernel lpo-vmlinuz

append console=ttyS0 console=tty0 ignore_loglevel,115200y8
  panic=1 initrd=lpo-initrd.gz rdinit=/linuxrc.sh
  printk.time=n file_caps virtio_net.gso=0
```

Impediments for RCA (ufo)

- Impossible to reproduce in CEE
- Probably impossible to reproduce in KVM simulator
- Tracing in guest kernel is not enough
- Tracing in host kernel is quite unfeasible
  - Different virtualized environments
  - Sometimes outside of our control

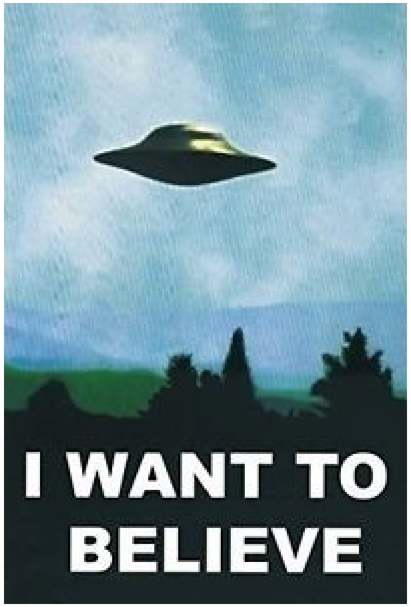
Unwanted side-effect (ufo)

- PL failed to boot in HP system



Workaround on top of workaround (ufo)

```
echo 0 > /proc/sys/net/bridge/bridge-nf-call-iptables
echo 0 > /proc/sys/net/bridge/bridge-nf-call-ip6tables
echo 0 > /proc/sys/net/bridge/bridge-nf-call-arptables
```



Case 5: Xfrm chessboard issue

- HU63960
- Standby crashes during catchup

Kernel panic (chess)

- Serious fault in kernel will reboot system

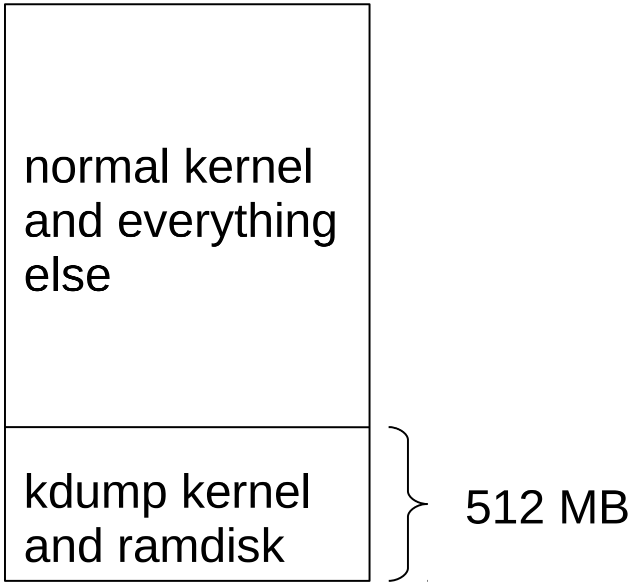
```
BUG: unable to handle kernel paging request at 00000000033b1bf0
IP: [<ffffffff81406b9a>] xfrm_hash_resize+0x11a/0x300
Oops: 0000 [#1] PREEMPT SMP
CPU 11
Modules linked in: authenc(N) esp6(N) [...]

Pid: 104, comm: kworker/11:1 Tainted: P
RIP: 0010:[<ffffffff81406b9a>]  [<ffffffff81406b9a>] xfrm_hash_resize+0x11a/0x300
RSP: 0018:ffff881015d59dc0  EFLAGS: 00010202
RAX: ffff880bab891c08 RBX: 00fff00000020000000 RCX: ffff880bab891c00
RDX: 000000000067631b RSI: 000000001fffffff RDI: 0000000000000000
[...]
Process kworker/11:1 (pid: 104, threadinfo ffff881015d58000, task ffff881015d560c0)
Stack: [...]
Call Trace:
 [<ffffffff81083743>] process_one_work+0x183/0x380
 [<ffffffff81085613>] worker_thread+0x183/0x430
 [<ffffffff81089c86>] kthread+0x96/0xb0
 [<ffffffff81448064>] kernel_thread_helper+0x4/0x10
 [<ffffffff81406b9a>] xfrm_hash_resize+0x11a/0x300
```

Kernel dump (chess)

- Added in IS-SBG 5.4
  - /flash/kerneldump
  - /flash/crash.dmesg
  - /home/homer/system/log/crash/crash.dmesg

Drawbacks with kerneldumps (chess)

- CONFIG_DEBUG_INFO and CONFIG_FRAME_POINTER makes kernel and modules larger



normal kernel and everything else

kdump kernel and ramdisk

512 MB

crash tool (chess)

- `crash lpo-vmlinux kerneldump`
- Kernel dump analysis tool
- Based on gdb with some extra commands but all standard gdb commands do not work
- The help command show only the extra commands (not the standard gdb commands)

bt -l (chess)

```
PID: 104    TASK: ffff881015d560c0  CPU: 11  COMMAND: "kworker/11:1"
 #0 [ffff881015d59970] machine_kexec at ffffffff8102ddda
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/kernel/machine_kexec_64.
c: 387
 #1 [ffff881015d599d0] crash_kexec at ffffffff810b5697
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/kernel/kexec.c: 1197
 #2 [ffff881015d59ab0] oops_end at ffffffff81440fb8
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/kernel/dumpstack.c: 321
 #3 [ffff881015d59ae0] no_context at ffffffff8103a7c9
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/mm/fault.c: 681
 #4 [ffff881015d59b20] __bad_area_nosemaphore at ffffffff8103a9e5
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/mm/fault.c: 750
 #5 [ffff881015d59bf0] bad_area_nosemaphore at ffffffff8103aa8e
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/mm/fault.c: 758
 #6 [ffff881015d59c00] do_page_fault at ffffffff8144308e
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/mm/fault.c: 1103
 #7 [ffff881015d59d10] page_fault at ffffffff8143fe75
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86_64/kernel/entry.S
    [exception RIP: xfrm_hash_resize+282]
    RIP: ffffffff81406b9a  RSP: ffff881015d59dc0  RFLAGS: 00010202
    RAX: ffff880bab891c08  RBX: 0000000020000000  RCX: ffff880bab891c00
    RDX: 000000000067631b  RSI: 000000001fffffff  RDI: 0000000000000000
    RBP: ffff881015d59e10   R8: 00000000033b1bf0   R9: 000000000067637c
    R10: 0000000000000010  R11: 0000000000000000  R12: 0000000000000010
    R13: ffffffff81a749b0  R14: ffffc900413d5000  R15: 0000000000000010
    ORIG_RAX: ffffffffffffffff  CS: 0010  SS: 0018
 #8 [ffff881015d59e18] process_one_work at ffffffff81083743
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/include/asm/atomic.h: 25
 #9 [ffff881015d59e68] worker_thread at ffffffff81085613
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 188
#10 [ffff881015d59ee8] kthread at ffffffff81089c86
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/kernel/kthread.c: 99
#11 [ffff881015d59f48] kernel_thread_helper at ffffffff81448064
    /local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/arch/x86/kernel/entry_64.S: 1191
```

sym command (chess)

```
crash> sym xfrm_hash_resize

ffffffff81401430 (t) xfrm_hash_resize
  [...]/net/xfrm/xfrm_policy.c: 486

ffffffff81406a80 (t) xfrm_hash_resize
  [...]/net/xfrm/xfrm_state.c: 108
```

dis -l (chess)

```
ffffffff81406a80 + 282 = ffffffff81406b9a

crash> dis -l ffffffff81406b9a 20
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 623
0xffffffff81406b9a <xfrm_hash_resize+282>:      mov    (%r8),%rdx
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 625
0xffffffff81406b9d <xfrm_hash_resize+285>:      test   %rdx,%rdx
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 624
0xffffffff81406ba0 <xfrm_hash_resize+288>:      mov    %rdx,0x8(%rcx)
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 625
0xffffffff81406ba4 <xfrm_hash_resize+292>:      je     0xffffffff81406bae <xfrm_hash_resize+302>
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 626
0xffffffff81406ba6 <xfrm_hash_resize+294>:      lea    0x8(%rcx),%rax
0xffffffff81406baa <xfrm_hash_resize+298>:      mov    %rax,0x8(%rdx)
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/include/linux/list.h: 627
0xffffffff81406bae <xfrm_hash_resize+302>:      mov    %rax,(%r8)
/local/scratch/etxjohc/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out/../linux/net/xfrm/xfrm_state.c: 86
0xffffffff81406bb1 <xfrm_hash_resize+305>:      movzwl 0xc4(%rcx),%eax
```

xfrm_hash_transfer function (chess)

```
static void xfrm_hash_transfer(struct hlist_head *list,
       struct hlist_head *ndsttable,
       struct hlist_head *nsrctable,
       struct hlist_head *nspitable,
       unsigned int nhashmask)
{
  hlist_for_each_entry_safe(x, entry, tmp, list, bydst) {
    unsigned int h;
    h = __xfrm_src_hash(&x->id.daddr, &x->props.saddr,
                        x->props.family, nhashmask);
    hlist_add_head(&x->bysrc, nsrctable+h);
}
```

Xfrm state hash table (chess)

- Stores Ipsec SA (Security Associations) used by IMS-AKA
- Starting size is 8 buckets
- Every insert checks how full hash table is and may double the size
- Maximum 1048576 buckets = 1Mbucket = 24MB

Try static table size (chess)

- Allocate what we need at start
- Max size would be too much when 500 VR (24MB x 500 = 12GB)
- Trace resizing to confirm behaviour
- Unexpectedly is went to 8 times the maximum (8Mbuckets = 192MB)

Back to the code (chess)

- What triggers resize?

```
static void xfrm_hash_grow_check(struct net *net,
                            int have_hash_collision)
{
   if (have_hash_collision &&
       (net->xfrm.state_hmask + 1) < xfrm_state_hashmax &&
       net->xfrm.state_num > net->xfrm.state_hmask)
      schedule_work(&net->xfrm.state_hash_work);
}
```

Back to crash (chess)

```
crash> print init_net.xfrm
$9 = {
  state_all = {
    next = 0xffff880d0376bca0,
    prev = 0xffff8810136ce8a0
  },
  state_bydst = 0xffffc900413d5000,
  state_bysrc = 0xffffc901987ee000,
  state_byspi = 0xffffc902187ef000,
  state_hmask = 268435455,  (256 Mbuckets x 24 = 6GB)
  [...]
```

Roll the dice (chess)

- The intended maximum is 24MB
- 1 extra doubling → 48MB
- 2 extra doublings → 96MB
- 3 extra doublings → 192MB
- 4 extra doublings → 384MB
- 5 extra doublings → 768MB
- 6 extra doublings → 1536MB
- 7 extra doublings → 3GB
- 8 extra doublings → 6GB
- 9 extra doublings → KABOOM!!!

To be continued (chess)

Trouble-shooting strategy

Major bugs are probably corrected in a later release

- Search with google or in linux repo
  - git log –grep=”whatever”
  - git log <filename>
- Very hard to find bugs by code inspection alone
- Code inspection can help to find search words

Information filter

- There are usually a lot of information and theories available
- Some are incorrect observations and a lot are incorrect assumptions
- Therefore don’t abandon a theory because it does not fit 100% with all “facts”
- Also bugs thwarts logic
- Verify uncertain assumptions with tracing whenever possible

Bisecting

- Tracing or other activities that aim to narrow off the problem area
  - It is usually not possible to investigate every theory
  - Prioritize theories that are easily testable (even unlikely ones)
  - Code review is valuable to get ideas

Hello world

- The C Programming Language

```
#include <stdio.h>

main()
{
  printf("hello world!\n");
}
```

Return code 13

```
#!/bin/bash

set –o errexit

hello

goodbye
```

Improvement

```
#include <stdio.h>

main()
{
  printf("hello kitty!\n");
  return 0;
}
```

Linux kernel

- v3.0 – 10 MLOC
- v3.12 – 12 MLOC
- v4.14 – 17 MLOC

Kernel and LPO practicalities

Kernel vs user space

- Limitations
  - No libc
    - printk instead of printf
    - Man section 9 instead of 2+3
  - No floating point operations

printk

- Prints to RAM log buffer and console
- Show log buffer with dmesg command
- Optional priority level
  - printk(KERN_WARN, format, …);
  - pr_warn(format, …);
- Priority filter controlled by sysctl kernel.printk
- Saved in ASI in log/oslogs/kernel-info and syslog plus dmesg

Rate limit printouts

```
if (net_ratelimit())
   printk(KERN_WARNING
          "dst cache overflow\n");
```

Counters

```
static int foo_counter;


foo_counter++;
```

Atomic integers

```
static atomic_t foo = ATOMIC_INIT(0);

atomic_inc(&foo);

atomic_read(&foo);
```

Proc interface

```
blade_0_5:~# cat /proc/sys/net/netfilter/nf_conntrack_frag6_high_thresh
4194304
```

Defining proc entries

net/ipv6/netfilter/nf_conntrack_reasm.c:

```
static struct ctl_table nf_ct_frag6_sysctl_table[] = {
{
  .procname= "nf_conntrack_frag6_high_thresh",
  .data   = &init_net.nf_frag.frags.high_thresh,
  .maxlen= sizeof(unsigned int),
  .mode   = 0644,
  .proc_handler  = proc_dointvec,
},
[...]
{ }
};
```

Add one entry

```
 {
       .procname  = "frag_count",
       .data  = NULL,
       .maxlen  = 0,
       .mode  = 0222,
       .proc_handler = frag_counters,
},
```

Entry function

```
static int frag_counters(struct ctl_table *ctl,
                         int write,
                         void __user *buffer,
                         size_t *lenp,
                         loff_t *fpos)
{
        printk("hello world\n");
        return 0;
}
```

```
# echo 1 > /proc/sys/net/netfilter/frag_count
```

netstat counters

- netstat -s
- netstat -s -A inet6
- Defined in net/ipv6/proc.c

```
SNMP_MIB_ITEM("Ip6ReasmFails",
  IPSTATS_MIB_REASMFAILS),
```

- Incremented with IP6_INC_STATS

```
IP6_INC_STATS(net, ip6_dst_idev(skb_dst(skb)),

          IPSTATS_MIB_REASMFAILS);
```

Adding a patch in LPO

```
cd lpo
make
cp –r out/tmp/kernel_build_dir/linux/
      /local/scratch/$USER/orig
cd /local/scratch/$USER
cp –r orig my_trace
```

Edit my_trace...

Add to makefile

- IS-SBG
  - **KERNEL_PATCHES** in lpo/lpo.mk
- vSBG
  - **SSP_PATCHES** in lpo/Makefile

Compile with patch

```
#!/bin/bash

set -ex
patch=my_trace
lpo=/local/scratch/$USER/is-sbg-lpo/src/syf/lpo/
cd /local/scratch/$USER
! diff -Naur -x TAGS orig $patch > $lpo/src/kernel/$patch.patch
cd $lpo
git add src/kernel/$patch.patch
git clean -fdx
make
cd out
mkdir -p ~/pub/$patch/src
cp lpo-bootfs.tar.gz lpo-initrd.gz lpo-vmlinuz ~/pub/$patch
cp $lpo/src/kernel/$patch.patch lpo-vmlinux ~/pub/$patch/src
```

Incremental compilation IS-SBG

```
cd lpo/out/tmp/kernel_build_dir/linux
```

```
make srctree=../linux -j8 O=/local/scratch/$USER/is-sbg-lpo/src/syf/lpo/out/tmp/kernel_build_dir/out bzImage
```

```
cp -r * /local/scratch/$USER/my_patch
```

ERICSSON