Subject: Re: [BRIDGE] Unaligned access on IA64 when comparing ethernet addresses
Posted by davem on Tue, 17 Apr 2007 21:09:41 GMT
View Forum Message <> Reply to Message

From: Stephen Hemminger <shemminger@linux-foundation.org>
Date: Tue, 17 Apr 2007 13:37:23 -0700

> The previous patch relied on the bridge id being aligned by
> the compiler (which happens as a side effect). So please use
> this instead.
>
> compare_ether_addr() implicitly requires that the addresses
> passed are 2-bytes aligned in memory.
>
> This is not true for br_stp_change_bridge_id() and
> br_stp_recalculate_bridge_id() in which one of the addresses
> is unsigned char *, and thus may not be 2-bytes aligned.
>
> Signed-off-by: Evgeny Kravtsunov <emkravts@openvz.org>
> Signed-off-by: Kirill Korotaev <dev@openvz.org>
> Signed-off-by: Pavel Emelianov <xemul@openvz.org>
> Signed-off-by: Stephen Hemminger <shemminger@linux-foundation.org>

bridge_id would be aligned by luck, because it is composed of char's
there is no explicit reason it should be aligned on at least an
unsigned short boundary.

I like the other patch much better, it provided explicit alignment and
is guarenteed to get rid of the problem.

Indeed, I wrote a test program on 32-bit Sparc to validate this:

```
struct bridge_id {
 unsigned char a[6];
 unsigned char b[6];
};

extern void bar(unsigned char *, unsigned char *);

void foo(void)
{
 unsigned char a;
 struct bridge_id b;

 bar(&b.a[0], &a);
}
```

foo() gets compiled like this:

```
foo:
 save %sp, -120, %sp
 add %fp, -21, %o0
 call bar, 0
  add %fp, -9, %o1
 jmp %i7+8
  restore
```

See? The bridge_id (passed in via %o0) is on an odd byte boundary on the stack.

So your patch isn't fixing the bug at all.

I'm going to apply the original patch, because that one will actually fix the problem and was actually tested on a system that saw the problem.
