Subject: [RFC] network namespaces
Posted by Andrey Savochkin on Tue, 15 Aug 2006 14:20:29 GMT
View Forum Message <> Reply to Message

Hi All,

I'd like to resurrect our discussion about network namespaces.
In our previous discussions it appeared that we have rather polar concepts
which seemed hard to reconcile.
Now I have an idea how to look at all discussed concepts to enable everyone's
usage scenario.

1. The most straightforward concept is complete separation of namespaces,
   covering device list, routing tables, netfilter tables, socket hashes, and
   everything else.

   On input path, each packet is tagged with namespace right from the
   place where it appears from a device, and is processed by each layer
   in the context of this namespace.
   Non-root namespaces communicate with the outside world in two ways: by
   owning hardware devices, or receiving packets forwarded them by their parent
   namespace via pass-through device.

   This complete separation of namespaces is very useful for at least two
   purposes:
   - allowing users to create and manage by their own various tunnels and
     VPNs, and
   - enabling easier and more straightforward live migration of groups of
     processes with their environment.

2. People expressed concerns that complete separation of namespaces
   may introduce an undesired overhead in certain usage scenarios.
   The overhead comes from packets traversing input path, then output path,
   then input path again in the destination namespace if root namespace
   acts as a router.

   So, we may introduce short-cuts, when input packet starts to be processes
   in one namespace, but changes it at some upper layer.
   The places where packet can change namespace are, for example:
   routing, post-routing netfilter hook, or even lookup in socket hash.

   The cleanest example among them is post-routing netfilter hook.
   Tagging of input packets there means that the packets is checked against
   root namespace's routing table, found to be local, and go directly to
   the socket hash lookup in the destination namespace.
   In this scheme the ability to change routing tables or netfilter rules on
   a per-namespace basis is traded for lower overhead.

> All other optimized schemes where input packets do not travel
> input-output-input paths in general case may be viewed as short-cuts in
> scheme (1). The remaining question is which exactly short-cuts make most
> sense, and how to make them consistent from the interface point of view.

My current idea is to reach some agreement on the basic concept, review
patches, and then move on to implementing feasible short-cuts.

Opinions?

Next in this thread are patches introducing namespaces to device list,
IPv4 routing, and socket hashes, and a pass-through device.
Patches are against 2.6.18-rc4-mm1.

Best regards,

Andrey
