CS522 TERM PROJECT

REPORT ON LEARNING LINUX NETWORKING

JIANHUA XIE

DEC. 6, 2000

Report on Learning Linux Networking

Jianhua Xie

1. GOAL

To learn the mechanisms of networking in Linux 2.4 kernel, including netfilter, NAT, traffic control, tunneling and how to use these mechanisms to implement load balancing, differentiated services and security control.

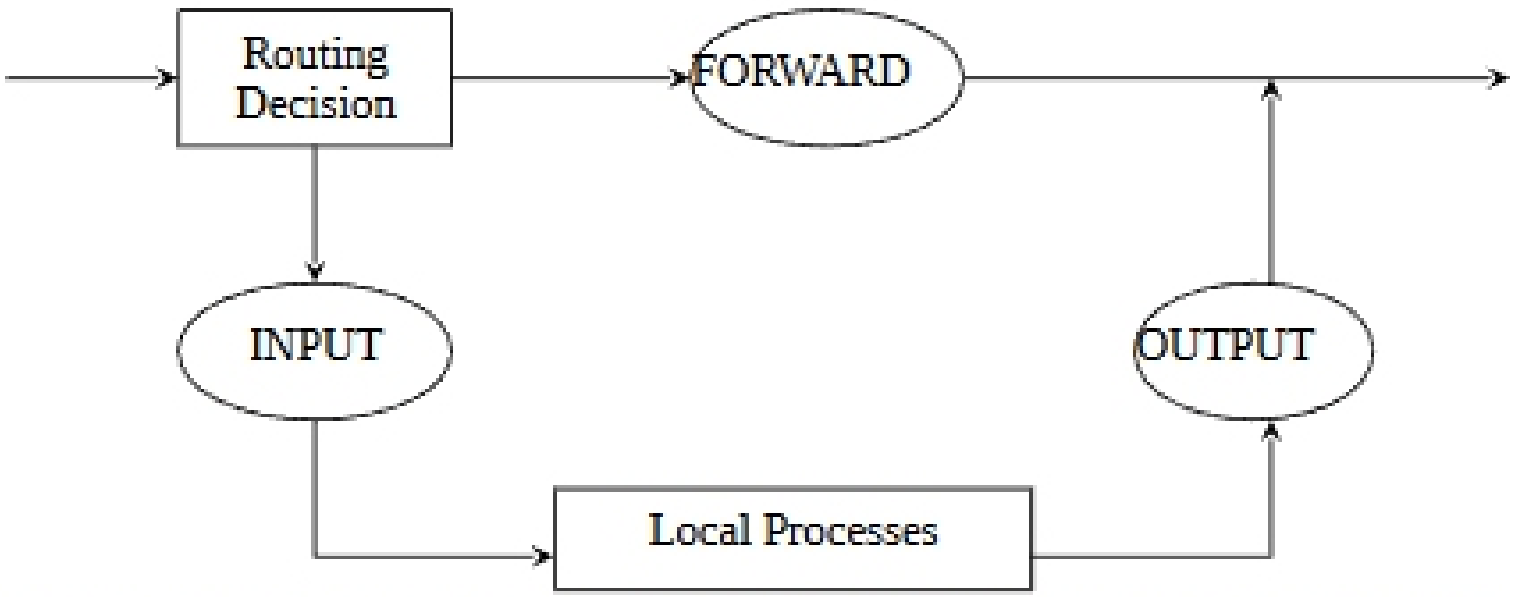
2. READING LIST

Linux 2.4 Advanced Routing HOWTO	by Gregory Maxwell etc.
Linux 2.4 Packet Filtering HOWTO: Introduction	by Russell
Linux 2.4 NAT HOWTO: Introduction	by Russell
Linux netfilter Hacking HOWTO	by Russell
Differentiated Services on Linux	by Werner Almesberger etc.
Definition of the Differentiated Services Field In the Ipv4 and Ipv6 Headers	by K. Nichols etc.
An Architecture for Differentiated Services	by S. Blake
The Linux Kernel (Chapter 10)	by David A. Rusling
Fast Firewall Implementations for Software-based Routers	by Lili Qiu etc.
Weighted Random Early Detection (WRED)	by Cisco
Cookie Tracking	by Chow
Link-sharing and Resource management Models For Packet Networks	by Sally Floyd, etc.
iproute2+tc notes	by dragon@snafu.freedom.org

3. NETFILTER IN LINUX

When we connect our computer systems or internal network to another network, such as Internet, we need some facilities to lend us the control on the connection to minimize the potential risks by allowing some kinds of traffic and disallowing others. Netfilter is one of these in Linux box. Basically, netfilter is a piece of software which looks into the header of packets as they pass through, and decides the fate of the entire packet. It might decide to discard the packet, or send the packet, or deliver the packet to user space for further processing.

The infrastructure of Linux netfilter is illustrated in the following graph.



There are three chains (3 ellipsis in the graph above) in the netfilter table: INPUT, OUTPUT, FORWARD. A chain is a checklist of rules. When a packet comes in, the kernel first looks at the destination (routing decision). If the packet is destined for the machine, it passed downwards in the diagram to the INPUT chain. Otherwise if the packet is destined to another machine and there is a rout to forward this packet, then it passes to FORWARD chain. When a local process generates a packet, the packet will pass to OUTPUT chain. In the netfilter chains, the packet will be check through against the rules defined in the chain, it one rule matches, the action (ACCEPT, DROP, or QUEQUE) with the matched rule will be activated against the packet.

With netfilter, we can specify which kind of traffic we are interested and which kind we want to discard. The user space tool for system administrator to add/delete rules to/from Linux netfilter chains is iptables. We'll discuss iptables tool in next section. But we will give some example of how to create filter rules to regulate access to your system and outside networks.

For example, if you have a single PPP connection (with interface name ppp0) to the Internet, and don't want anyone coming back into your system, you can achieve this by the following configuration (before doing this, we should assure that ip_conntrack and ip_conntrack_ftp modules have been installed, this is done by command 'insmod' with module name as parameter):

```
# create a new netfilter chain to block any connections from outside world
>iptables –N block
>iptables –A block –m state –state ESTABLISHED, RELATED –j ACCEPT
>iptables –A block –m state –state NEW –I !ppp0 –j ACCEPT
>iptables –A block –j DROP
```