Virtual Switching Technologies

Or: How I Learned To Stop Worrying And Love ARP

Adrian Gschwend

netlabs.org - Open Source Software

Warpstock Europe 2007, Valkenswaard, Netherlands

Outline

Ethernet

- Used in LANs
- IEEE 802.3 Tagged MAC Frame (Ethernet-II)
- Collision Detection (CSMA/CD)
- Wired or Wireless (IEEE 802.11)
- Currently up to 10GBit/sec on copper
- Also on fiber, rarely to the desk
- Physical Layer (OSI Layer 1)

ARP

- Address Resolution Protocol
- Primarly used for IP
- MAC-Address (Media Acccess Control)
- Vital for every IP stack
- Host-to-network layer

The ARP Cache

```
1 playground:~ ktk$ arp -a
2 fritz.fonwlan.box (192.168.1.1) at 0:1a:4f:54:9:27 on ↩
      en0 [ethernet]
3 trancentral (192.168.1.23) at 0:6:5b:27:b8:73 on en0 [↩
      ethernet]
```

Routing and ARP

- Request to the Internet from Layer 7 (Application)
- Stack generates packet
- Stack decides where to send it (routing table, netmask)
- Outside of LAN, specific gateway
- For Internet mostly default gateways
- What is the MAC address of the default gateway?

Resolving an IP with ARP

tcpdump shows what happens:

```
1 ktk@trancentral:~$ sudo tcpdump -n -i ath0 "arp"
2 tcpdump: verbose output suppressed, use -v or -vv for ↩
      full protocol decode
3 listening on ath0, link-type EN10MB (Ethernet), capture↩
      size 96 bytes
4 arp who-has 192.168.2.1 tell 192.168.2.200
5 arp reply 192.168.2.1 is-at 00:1a:4f:54:09:27
```

Conclusion

- Ethernet is mainly for local networks (LAN)
- ARP is vital for IP to MAC translation
- Every network device has its MAC address

Ethernet Hub

- Connects multiple twisted pair/fiber optic Ethernet devices
- Makes them act as a single segment
- Multiport repeater
- Not very intelligent (all to all, broadcasting)

Ethernet Switch

- Same purpose as hubs
- Knows which MAC on which port
- Thus no repeating on each port
- Replaced hubs mostly

Router

- Connects LAN to WAN
- Connects different subnets
- Selects best path for a packet
- Layer 3 switch (Network Layer)
- Not part of this presentation :)

Network Bridge

- Connects multiple network segments
- Layer 2 switch (Data-Link Layer)
- On Ethernet MAC translations over a WAN
- Tunnel Ethernet segment to another place

Virtual Switch

- Same purpose as a Ethernet switch
- Implemented in kernel of the OS
- Device Driver
- TUN/TAP on most Unix implementations (API)

TUN device

- Network TUNel
- Simulates network layer device
- Layer 3 (IP packets)
- Just IP based protocols
- Used with routing
- Sample device: PPP interface

TAP device

- A network TAP
- Simulates Ethernet device
- Layer 2 (Ethernet frames)
- Any protocol on top
- Used with network bridges
- Sample device: Virtual network card

TUN on eComStation

- Provided by the operating system
- Used by PPP devices
- Point to point interface (no ARP)
- Only one target on the other side
- Used by the current OpenVPN port

TAP on eComStation

- Not provided by the OS
- Implemented by Willibald Meyer in `PROT.OS2`
- Can be used in many ways

PROT.OS2

- Implements TAP device
- Implements sniffing interface
- Implements Virtual Switch
- Installed via MPTN
- Appears as TAP$ - TAP8$

PROT.OS2 as TAP

- Virtual NIC
- Accepts any protocol
- IOCTL for read/write
- From protocol stack to application or vice versa
- Virtual MAC address (can be set)
- ARP handling done by driver
- Can be turned off (PROXYARP="OFF")
- Used by VPNC, OpenVPN, eCSConet

PROT.OS2 as Sniffer

- PROT.OS2 hooks into *PROTMAN*
- Gets all requests from protocol to NIC or vice versa
- Copies packets into a trace-buffer
- IOCTL interface for reading from buffer
- Can be read out in application level
- Implemented in ETHR.EXE

PROT.OS2 as Virtual Switch

- PROT.OS2 hooks into *PROTMAN*
- Gets all requests from protocol to NIC or vice versa
- Default: sniffing mode
- Virtual Switch:
  - VirtualBox attachs to TAP$
  - `TAPSWCH TAP$ connect lan0`

Behind the Scene

- Enabling the switch kills direct connections in *PROTMAN*
- The physical NIC switches to promiscous mode
- Proxyarp for TAP gets disabled
- Requests from the protocol assigned to TAP get ignored
- Three possible packet sources:
  - Host OS stack
  - Guest OS stack
  - Packets from the LAN (via physical NIC)

Sample Usage

- Create a TAP interface (via MPTN)
- Bind to physical interface
- Assign virtual machine to it
- Use it

Outlook

- Very flexible because of *PROTMAN* hooks
- Firewalling would be possible too
- Sniffing with *libpcap*
- Other ideas?

Questions/Feedback

Any questions?