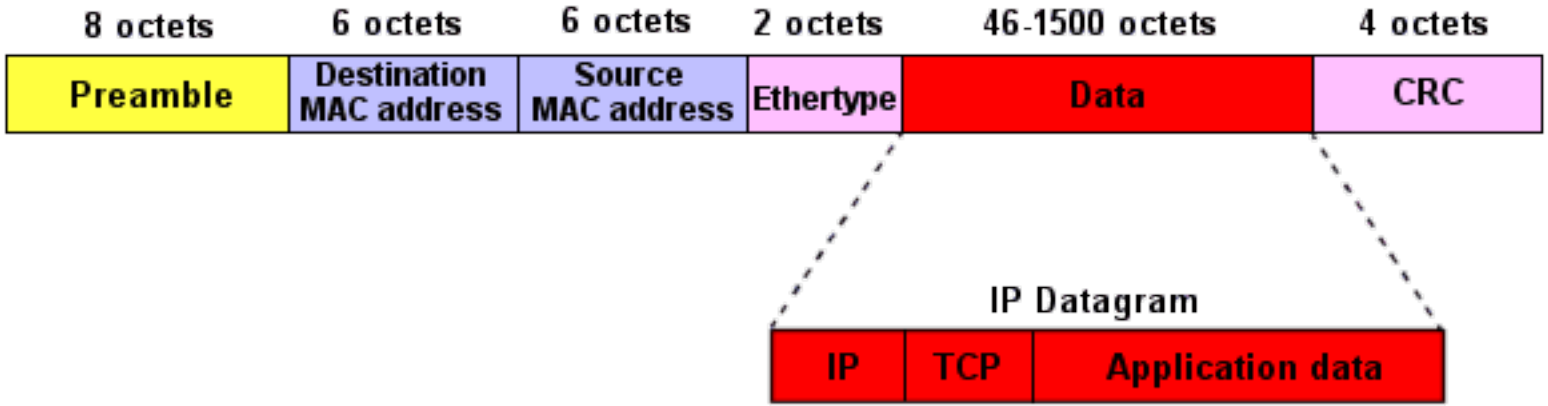
IPv6 and MTU

battlemesh v7

For who is this talk ?

MTU Maximum Transmission Unit

MTU is the max size in bytes we can deliver to the L2 layer interface without fragmenting IP packets.



Tipical values for MTU

Tipical values are:

ethernet	**1500**
ethernet jumbo	**9198**
802.11	**7981**

NOC life: how to troubleshoot ?

Wrong approach to ping

ping -s 1500 <x.x.x.x>

what is wrong ?

the 1500 makes no sense

there is not indication about fragmenting

ping

ping -s 1472 -M do <x.x.x.x>

for testing a ethernet of 1500 bytes MTU

do means: do prohibit fragmentation

1472	ICMP payload
8	ICMP header
20	IPv4 header

ping: try to go above the limit

Easily try to ping a host in your LAN exceeding the MTU size:

ping -s 1473 -M do <x.x.x.x>

From localhost (192.168.1.00) icmp_seq=1
Frag needed and DF set (mtu = 1500)

ping6

ping -s 1452 -M do <x.x.x.x>

for testing a ethernet of 1500 bytes MTU

do means: do prohibit fragmentation

1452	ICMP payload
8	ICMPv6 header
40	IPv6 header

ping6: try to go above the limit

Easily try to ping a host in your LAN exceeding the MTU size:

ping -s 1453 -M do <2001::xx>

From 2001::xx

icmp_seq=1 Packet too big: mtu=1500

Check IPv4 MTU in use

$ ip addr show dev eth0| grep mtu

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 1500 qdisc pfifo_fast state UP qlen 1000

you can set the MTU with iproute

Check IPv6 MTU in use

```
cubie@Cubian:~$ sysctl net.ipv6.conf.eth0.mtu
net.ipv6.conf.eth0.mtu = 1496
cubie@Cubian:~$
```

radvd

```
interface eth0
{
    AdvLinkMTU 1496;
    AdvSendAdvert on;
    AdvManagedFlag off;
    AdvOtherConfigFlag off;

    prefix 2001:xxxx:xxxx:xxxx::/64
    {
            AdvOnLink on;
            AdvAutonomous on;
            AdvRouterAddr on;
    };
};
```

IPv6 blackholes issue

IPv6 No fragmentation is possible !

IPv4 will fragment so we just have a performance issue

If ICMPv6 is filtered, PMTU discovery will not work. Easily routers will blackhole traffic that is too big.

Usually fat traffic is in replies (think of HTTP web pages will not load completely)

iptables

Good rule to know:

iptables -A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

This will intercept TCP sessions and adjust the MSS to fit with the MTU of the output interface

ip6tables

Who will implement this ?

Questions ?

feel free to write to me

zioproto@gmail.com

http://twitter.com/zioproto