Networking Basics

04e - PING considered harmful (for network monitoring)

Wolfgang Tremmel
academy@de-cix.net

Networking Basics

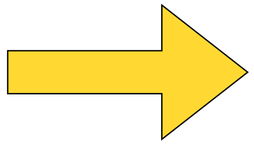
DE-CIX Academy

01 - Networks, Packets, and Protocols

02 - Ethernet, 02a - VLANs, 02b - QinQ

03 - IP, 03a - Routing, 03b - Global routing

04a - UDP, 04b - TCP, 04c - ICMP, 04d - Traceroute



04e - Ping considered harmful (for network monitoring)

05 - Uni-, Broad-, Multi-, and Anycast

06a - Domain Name System (DNS)

07a - SMTP, 07b - HTTP

DE-CIX

Recently in customer service....

Based on a true story

About PING

About PING

Most of you might have done this before....

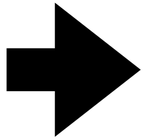
- You *ping* a system to check if it is "alive"
  - "Alive" in this case means reachable over the Internet
- So *ping* must use a mechanism to send a packet
  - and to process a reply
- The packets sent and received are **ICMP** messages
  - called "Echo Request" and "Echo Reply"

DE CIX

Internet Model

ICMP - Internet Control Message Protocol

- ICMP uses IP for transport
  - So it is "above" the Internet layer
- But it does not have anything "on top" of it
- This is true for both IPv4 and IPv6



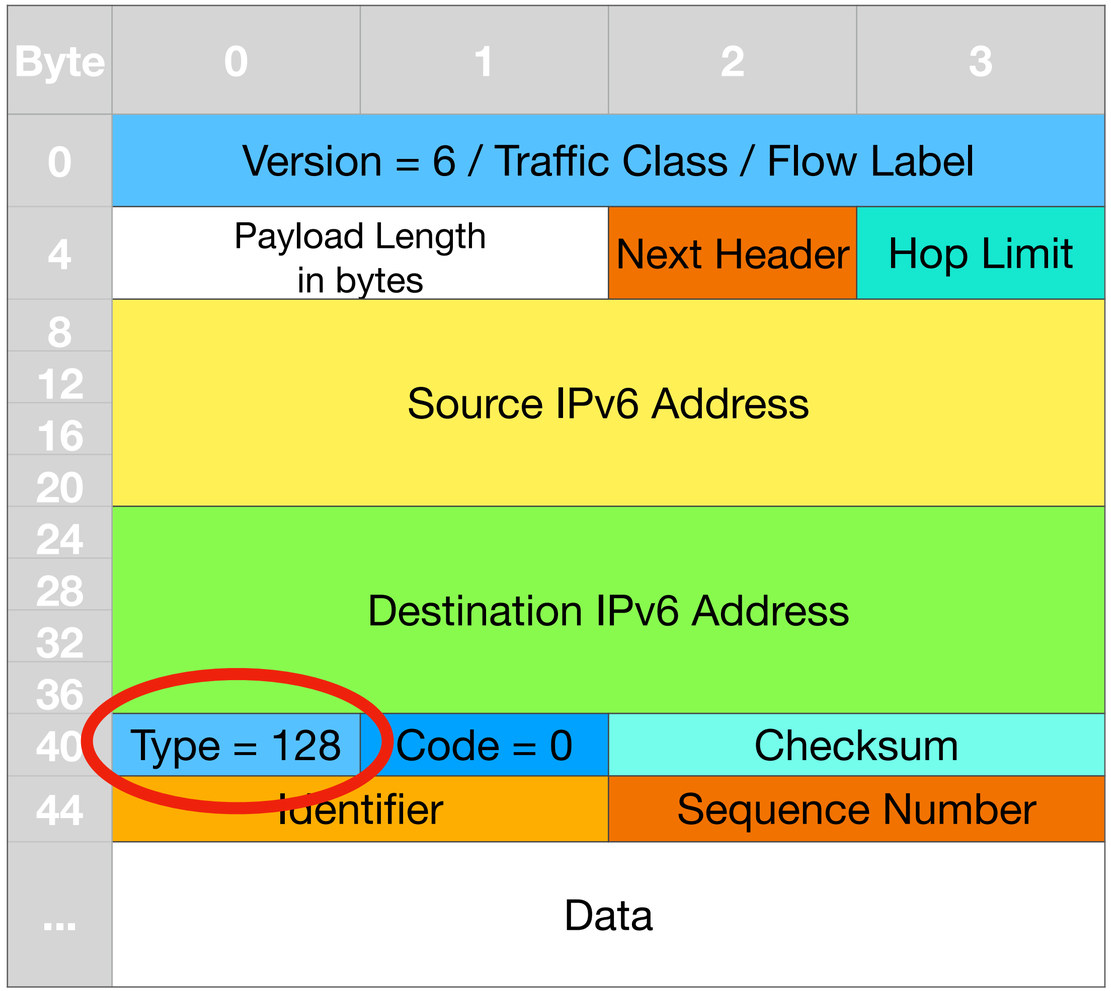
Layer	Name
4	**ICMP**
3	Internet
2	Link
1	Physical

DE CIX

ICMP Echo Request

For IPv6

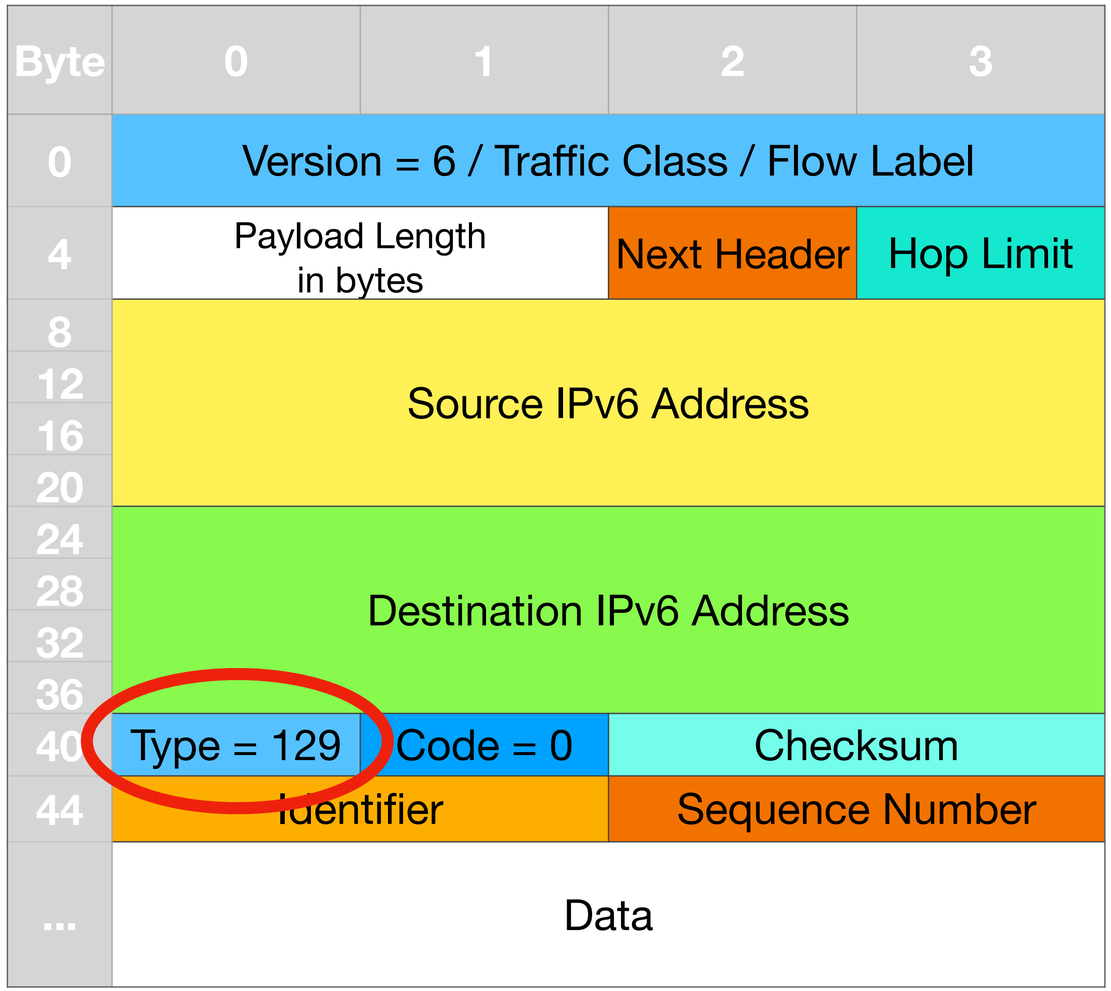
- ICMP Type for Echo Request in IPv6 is 128
- Code is always zero
- Checksum covers the ICMP part and (in IPv6 only parts of the IPv6 header)
- Identifier is a 16bit number (may be zero) to match replies to requests
- Sequence number is a 16 bit counter also to match replies to request.
  - Usually counts up for each ping
- Data can be any data to make the packet larger



ICMP Echo Reply

For IPv6

- ICMP Type for Echo Reply in IPv6 is 129
- Code is always zero
- Checksum - calculated the same way as in Echo Request
- Identifier is a 16bit number (may be zero) copied from Echo Request
- Sequence number is a 16 bit counter copied from Echo Request.
- Data is also copied from Echo Request



DE CIX

Processing a PING

Echo Request

Byte	0	1	2	3
0	Version | Header Length	DSCP / ECN	Total Length 20..65535	
4	Identification		Flags / Fragment Offset	
8	Time To Live	Protocol	Header Checksum	
12	Source IPv4 Address: 192.0.2.1			
16	Destination IPv4 Address: 198.51.100.34			
40	Type = 8	Code = 0	Checksum	
44	Identifier 0		Sequence Number 22	
...	Data Some Data			

DE CIX

Processing a PING

Echo Request

Byte	0	1	2	3
0	Version | Header Length	DSCP / ECN	Total Length 20..65535	
4	Identification		Flags / Fragment Offset	
8	Time To Live	Protocol	Header Checksum	
12	Source IPv4 Address: 192.0.2.1			
16	Destination IPv4 Address: 198.51.100.34			
40	Type = 8	Code = 0	Checksum	
44	Identifier 0		Sequence Number 22	
...	Data Some Data			

Echo Reply

Byte	0	1	2	3
0	Version | Header Length	DSCP / ECN	Total Length 20..65535	
4	Identification		Flags / Fragment Offset	
8	Time To Live	Protocol	Header Checksum	
12	Source IPv4 Address:			
16	Destination IPv4 Address:			
40	Type = 0	Code = 0	Checksum	
44	Identifier		Sequence Number	
...	Data			

DE CIX

Processing a PING

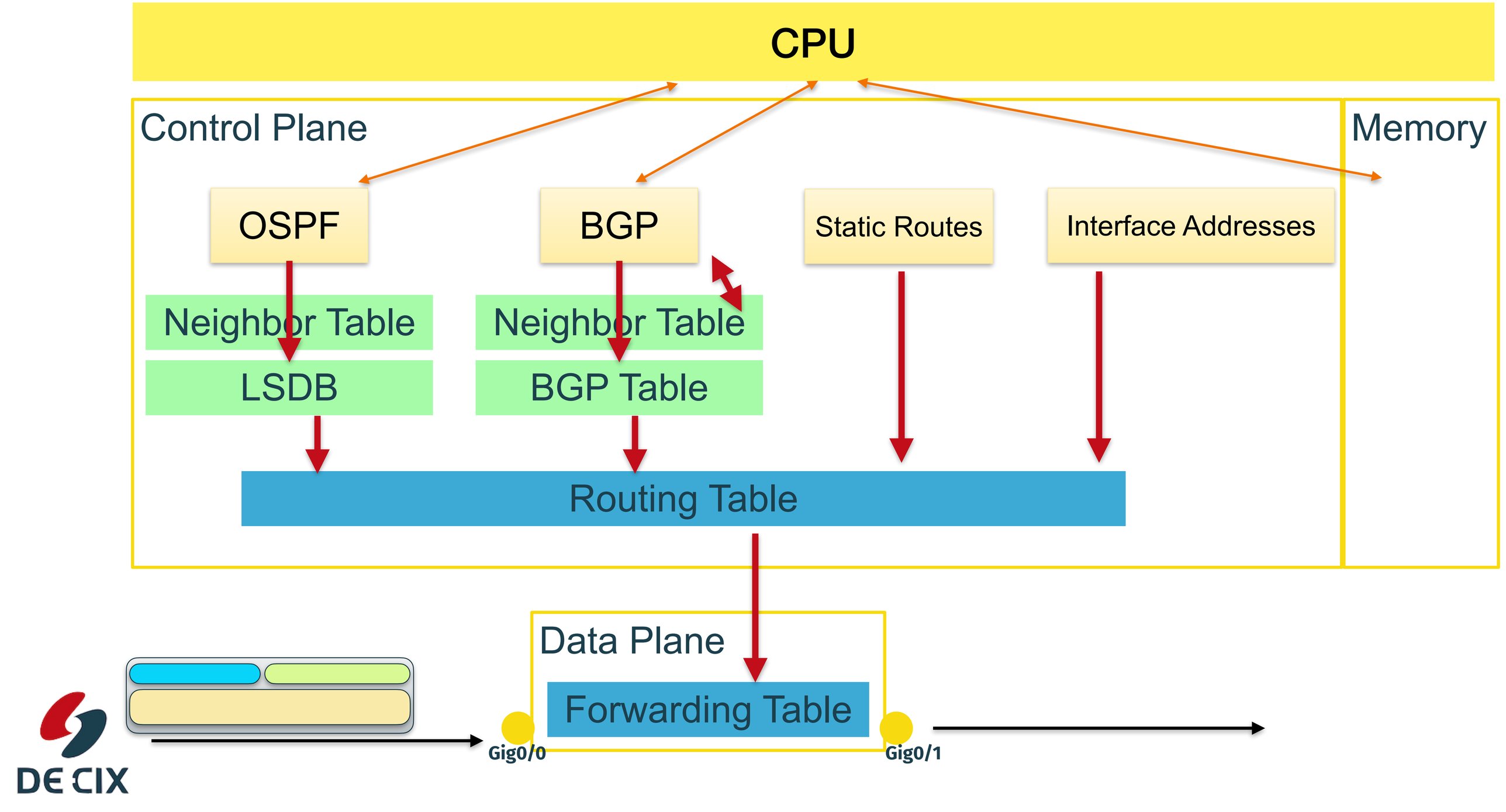
- Replying to a Ping is not "for free"
- It costs CPU cycles
- The reply packet needs to be generated
  - and the entries filled
  - with information from the request
- All this has to be done by the CPU

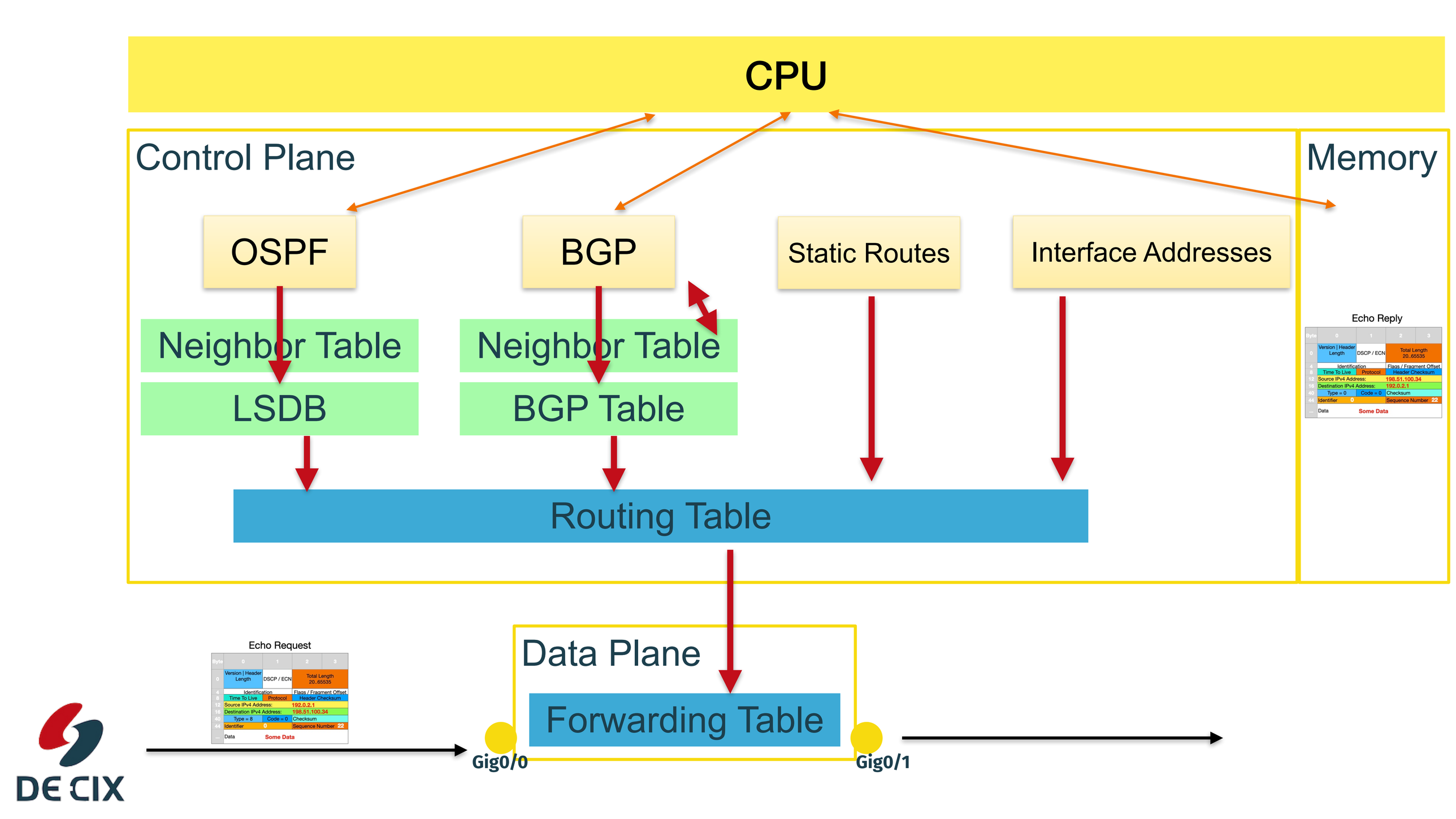
Echo Reply

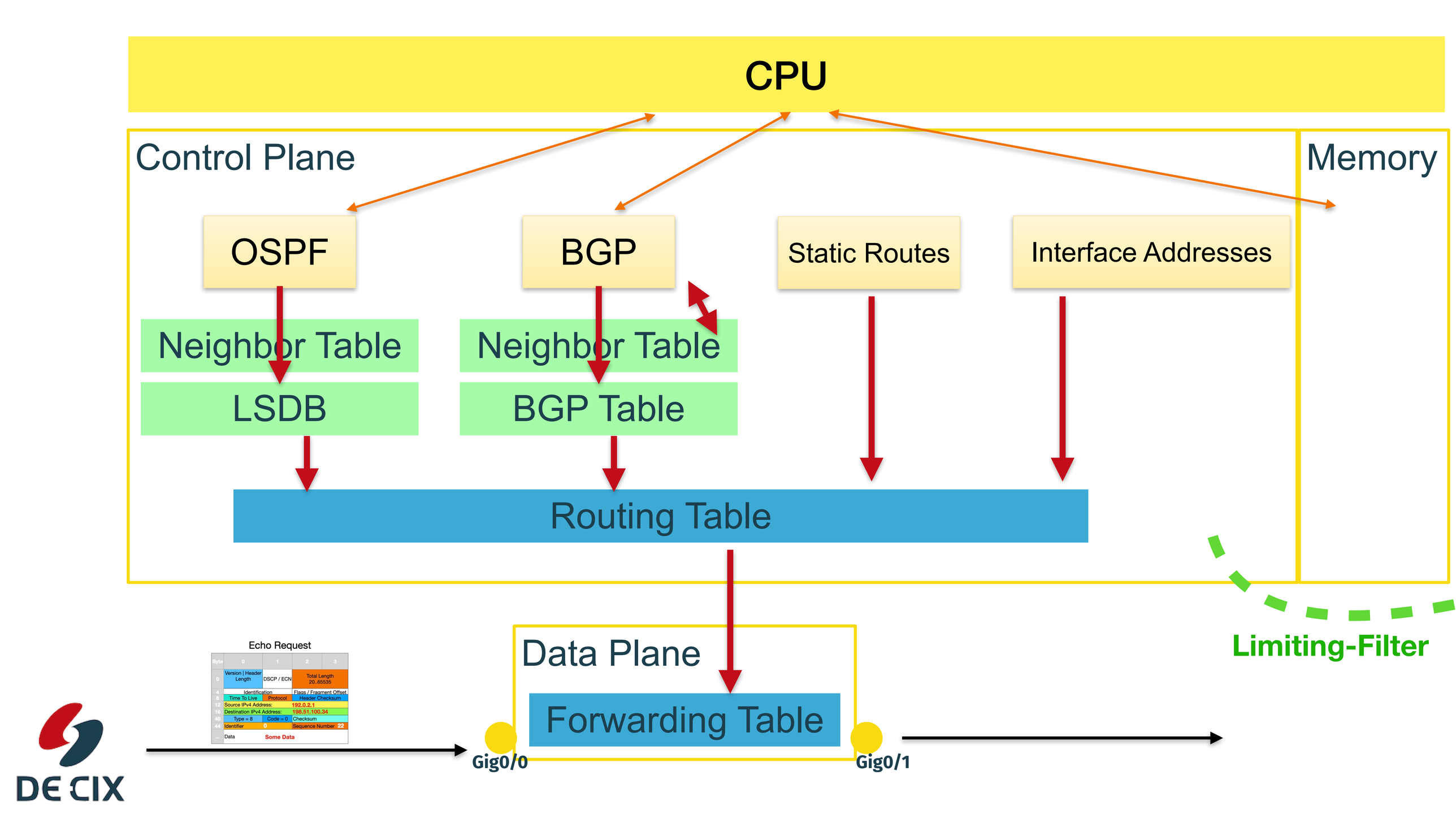
Byte	0	1	2	3
0	Version | Header Length	DSCP / ECN	Total Length 20..65535	
4	Identification		Flags / Fragment Offset	
8	Time To Live	Protocol	Header Checksum	
12	Source IPv4 Address: 198.51.100.34			
16	Destination IPv4 Address: 192.0.2.1			
40	Type = 0	Code = 0	Checksum	
44	Identifier 0		Sequence Number 22	
...	Data Some Data			

DE CIX

PINGing a router


CPU
Control Plane
Memory
OSPF
BGP
Static Routes
Interface Addresses
Neighbor Table
LSDB
Neighbor Table
BGP Table
Routing Table
Data Plane
Forwarding Table
Gig0/0
Gig0/1

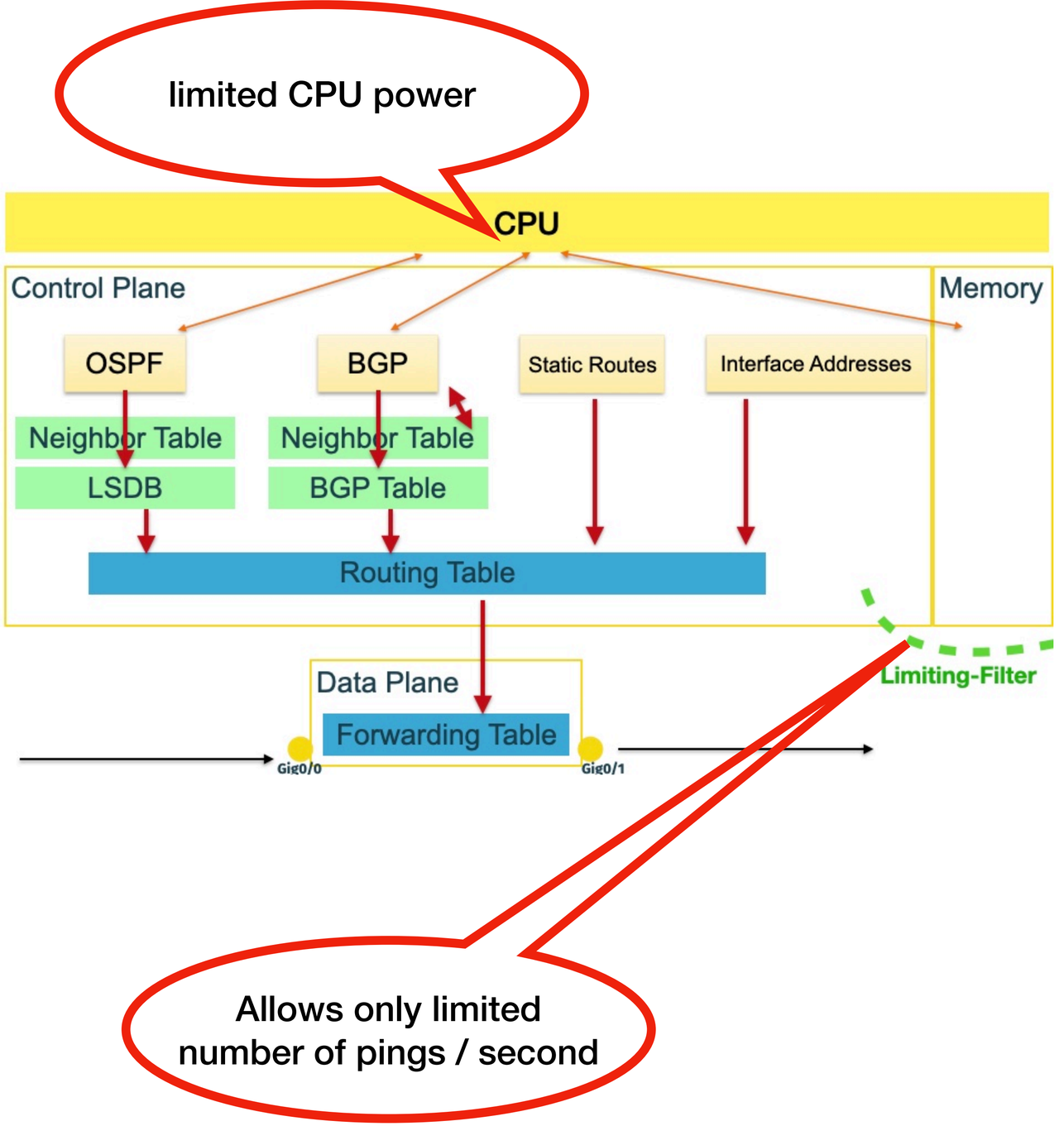

CPU
Control Plane
Memory
OSPF
BGP
Static Routes
Interface Addresses
Neighbor Table
LSDB
Neighbor Table
BGP Table
Routing Table
Echo Reply
Byte 0 1 2 3
0 Version | Header Length DSCP / ECN Total Length 20..65535
4 Identification Flags / Fragment Offset
8 Time To Live Protocol Header Checksum
12 Source IPv4 Address: 198.51.100.34
16 Destination IPv4 Address: 192.0.2.1
40 Type = 0 Code = 0 Checksum
44 Identifier 0 Sequence Number 22
... Data Some Data
Data Plane
Forwarding Table
Gig0/0
Gig0/1
Echo Request
Byte 0 1 2 3
0 Version | Header Length DSCP / ECN Total Length 20..65535
4 Identification Flags / Fragment Offset
8 Time To Live Protocol Header Checksum
12 Source IPv4 Address: 192.0.2.1
16 Destination IPv4 Address: 198.51.100.34
40 Type = 8 Code = 0 Checksum
44 Identifier 0 Sequence Number 22
... Data Some Data


CPU
Control Plane
Memory
OSPF
BGP
Static Routes
Interface Addresses
Neighbor Table
Neighbor Table
LSDB
BGP Table
Routing Table
Limiting-Filter
Data Plane
Forwarding Table
Gig0/0
Gig0/1
Echo Request
Byte
0
1
2
3
Version | Header Length
DSCP / ECN
Total Length 20..65535
Identification
Flags / Fragment Offset
Time To Live
Protocol
Header Checksum
Source IPv4 Address:
192.0.2.1
Destination IPv4 Address:
198.51.100.34
Type = 8
Code = 0
Checksum
Identifier
0
Sequence Number
22
Data
Some Data

Protecting the Router

By limiting CPU access

- A few pings are not a problem
- But if millions and millions of pings need to be processed, this can overload the CPU
- So a limiting filter or *policer* may regulate CPU access
- If a router has no limiter, the CPU is the limit, pings might still be dropped
- Dropped pings *appear* like packet loss

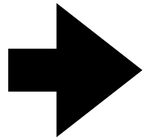


DE CIX

Conclusion

Conclusion

PING towards routers



Layer	Name
4	ICMP
3	Internet
2	Link
1	Physical

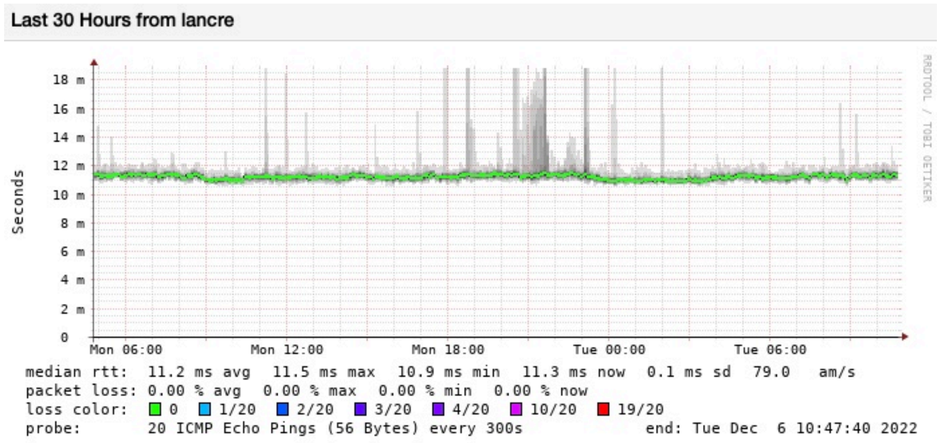
- ICMP packets to a router are processed by the routers CPU
- The access to the CPU may be limited by a policer
- Policed ping request appear like packet loss (but they are not)
  - Same if the CPU is overloaded
- You should not flood-ping other entities routers without their consent
  - Doing so might be seen as a DOS attack

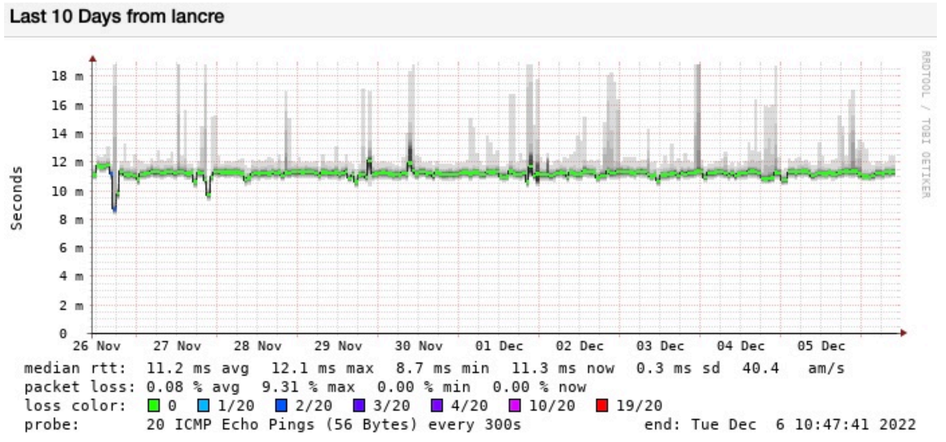
DE CIX

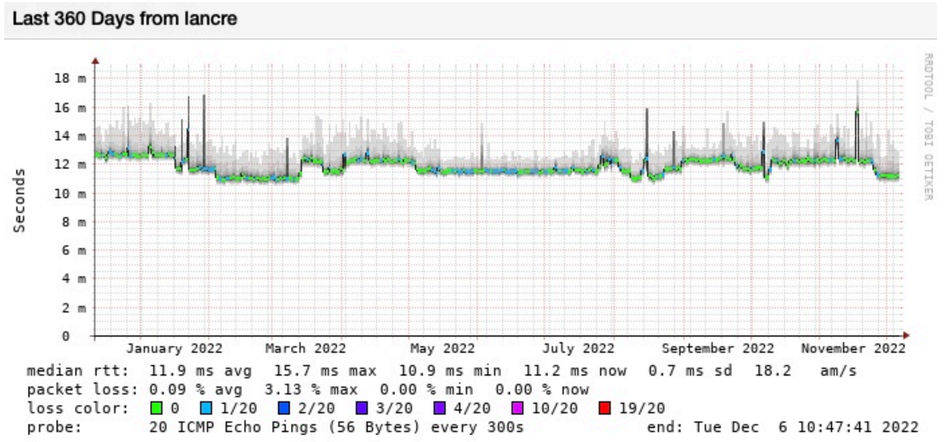
But... how can I monitor?

Using PING for monitoring

- Wait - did you not say ping is bad?
- I said *massive* ping is bad
- Smokeping per default sends 20 pings every 5 minutes which is IMHO perfectly ok
- Results are shown in a nice graphic







iPerf3

Performance testing

- If you want to test performance with a lot of packets
- **Never target a router**
- Ask the network you want to test with for starting *iPerf3* on a linux server
- And test against this server
- iPerf3 is an open source performance tester

Client:

```
Client:home wtremmel$ iperf -V -c server.example.com
------------------------------------------------------------
Client connecting to server.example.com, TCP port 5001
TCP window size:  129 KByte (default)
------------------------------------------------------------
[  6] local 198.51.100.1 port 63951 connected with 192.0.2.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  6]  0.0-10.0 sec   719 MBytes   603 Mbits/sec
```

Server:

```
root@example /var/log # iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 192.0.2.1 port 5001 connected with 198.51.100.1 port 28949
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0255 sec   835 MBytes   699 Mbits/sec
[  2] local 192.0.2.1 port 5001 connected with 198.51.100.1 port 29172
[ ID] Interval       Transfer     Bandwidth
[  2] 0.0000-10.0238 sec   719 MBytes   601 Mbits/sec
```

DE CIX

Thank you!

academy@de-cix.net

DE-CIX Management GmbH | Lindleystr. 12 | 60314 Frankfurt | Germany
Phone + 49 69 1730 902 0 | sales@de-cix.net | www.de-cix.net

Links and further reading

Links and further reading

- Internet protocol - https://en.wikipedia.org/wiki/Internet_Protocol
- Protocol stack - https://en.wikipedia.org/wiki/Protocol_stack
  - Transport Layer: https://en.wikipedia.org/wiki/Transport_layer
  - Datagram: https://en.wikipedia.org/wiki/Datagram
- IP Network Model: https://en.wikipedia.org/wiki/Internet_protocol_suite
- IPv4
  - IPv4 - https://en.wikipedia.org/wiki/IPv4
- IPv6
  - IPv6 itself - https://en.wikipedia.org/wiki/IPv6
  - IPv6 header - https://en.wikipedia.org/wiki/IPv6_packet
- History of Internet and IP
  - Internet Hall of Fame - https://internethalloffame.org
  - Defense Advanced Research Projects Agency (DARPA) - https://www.darpa.mil
  - ARPANET - https://www.darpa.mil/about-us/timeline/arpanet
  - The "Protocol Wars" - https://en.wikipedia.org/wiki/Protocol_Wars

DE CIX

Links and further reading

- ICMPv4:
  - Wikipedia: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
  - Definition in RFC792
  - Depreciation of some ICMP types: RFC6918
  - IANA list of ICMP types and codes: http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml
- ICMPv6:
  - Wikipedia: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_for_IPv6
  - Definition in RFC4443 (first definition, now obsolete, was in RFC1885)
    - Echo Request and Echo Reply: https://www.rfc-editor.org/rfc/rfc4443#section-4
  - IANA list of ICMPv6 types and codes: https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml
- Smokeping: https://oss.oetiker.ch/smokeping/
- iPerf3: https://software.es.net/iperf/

DE CIX

Internet RFCs (Standards)

- There are too many RFCs dealing with IPv4 and IPv6 to be listed here
- Just go to https://tools.ietf.org/html/ and use the search field
- How does something become RFC? https://www.rfc-editor.org/pubprocess/
- The IETF - Internet Engineering Task Force

DE CIX