An introduction to

Wireshark / Tshark

It provides a user-friendly graphical interface, but it also has a command-line version called "TShark".

Jiaxuanli4@link.cuhk.edu.cn

What is Wireshark?

- Wireshark is a **network packet analyzer**.

  A network packet analyzer presents **captured packet data** in as much detail as possible.

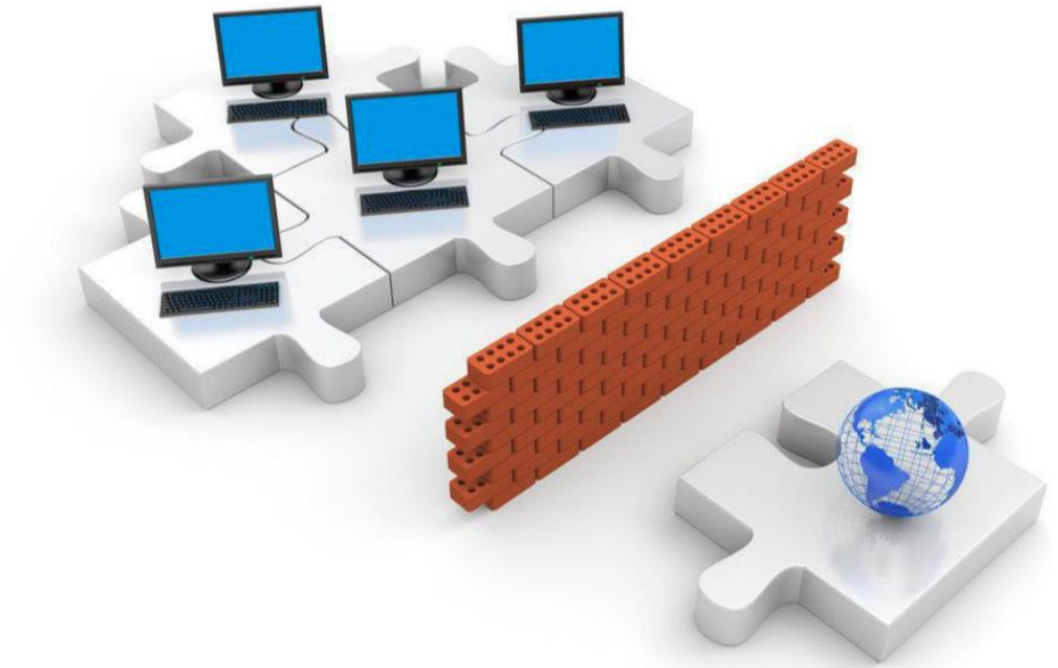
  It works on **various platforms** such as Windows, Linux, and Mac.

- You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable.
- In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.

What can wireshark do?

1. **Network troubleshooting**: Wireshark can capture network packets and display detailed packet information, helping to analyze the root cause of network issues.

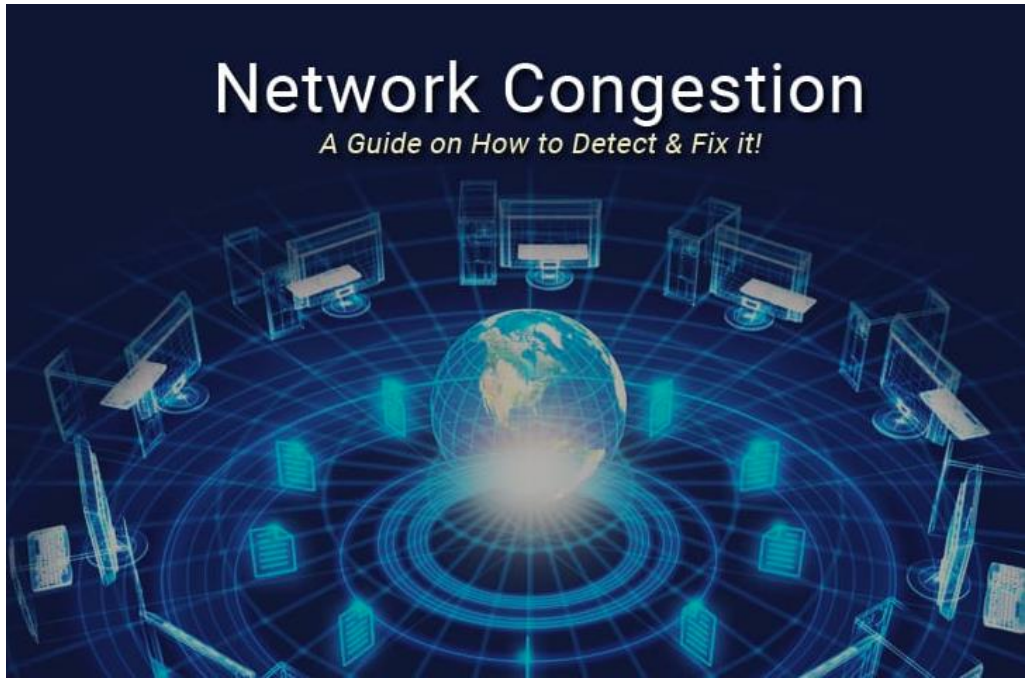
Network administrators



What can wireshark do?

1. **Network troubleshooting**: Wireshark can capture network packets and display detailed packet information, helping to analyze the root cause of network issues.
2. **Network performance optimization**: Wireshark can analyze network traffic patterns, anomalies, and performance problems, enabling network administrators to optimize network performance.

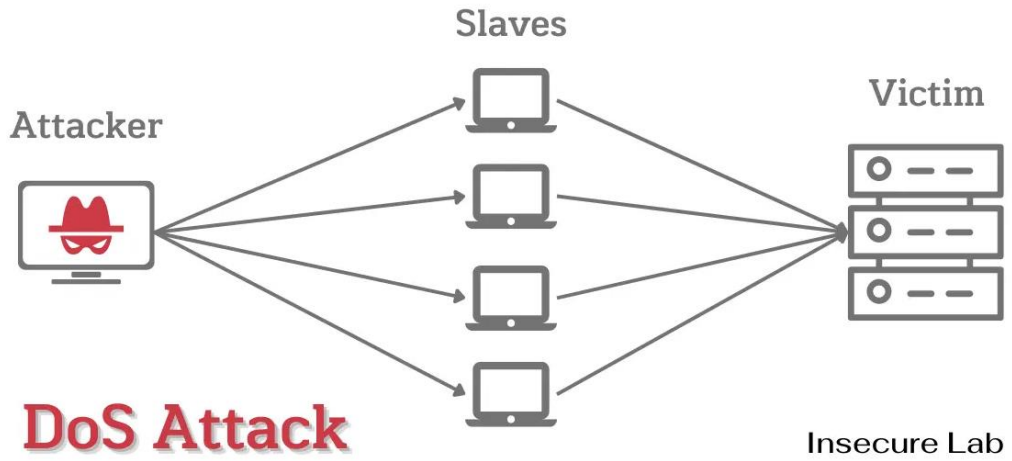
QA engineers



What can wireshark do?

1. **Network troubleshooting**: Wireshark can capture network packets and display detailed packet information, helping to analyze the root cause of network issues.
2. **Network performance optimization**: Wireshark can analyze network traffic patterns, anomalies, and performance problems, enabling network administrators to optimize network performance.
3. **Network security analysis**: Wireshark can assist in detecting and analyzing network attacks, such as denial-of-service attacks or malware propagation, by examining packet contents and patterns.

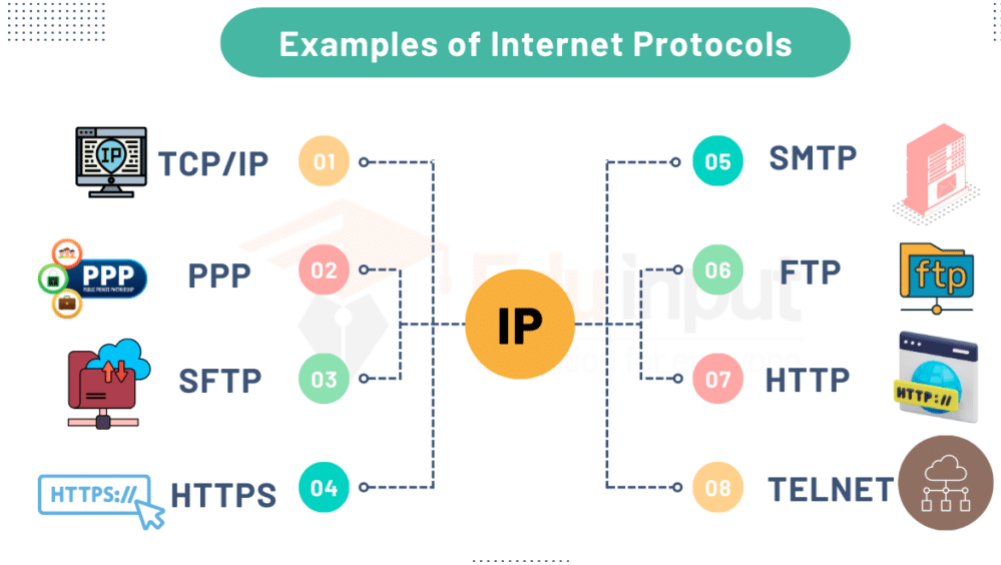
Network security engineers



What can wireshark do?

1. **Network troubleshooting**: Wireshark can capture network packets and display detailed packet information, helping to analyze the root cause of network issues.
2. **Network performance optimization**: Wireshark can analyze network traffic patterns, anomalies, and performance problems, enabling network administrators to optimize network performance.
3. **Network security analysis**: Wireshark can assist in detecting and analyzing network attacks, such as denial-of-service attacks or malware propagation, by examining packet contents and patterns.
4. **Protocol development and debugging**: Wireshark supports various protocol decoders and can be used for developing and debugging network protocols by inspecting packet-level details.

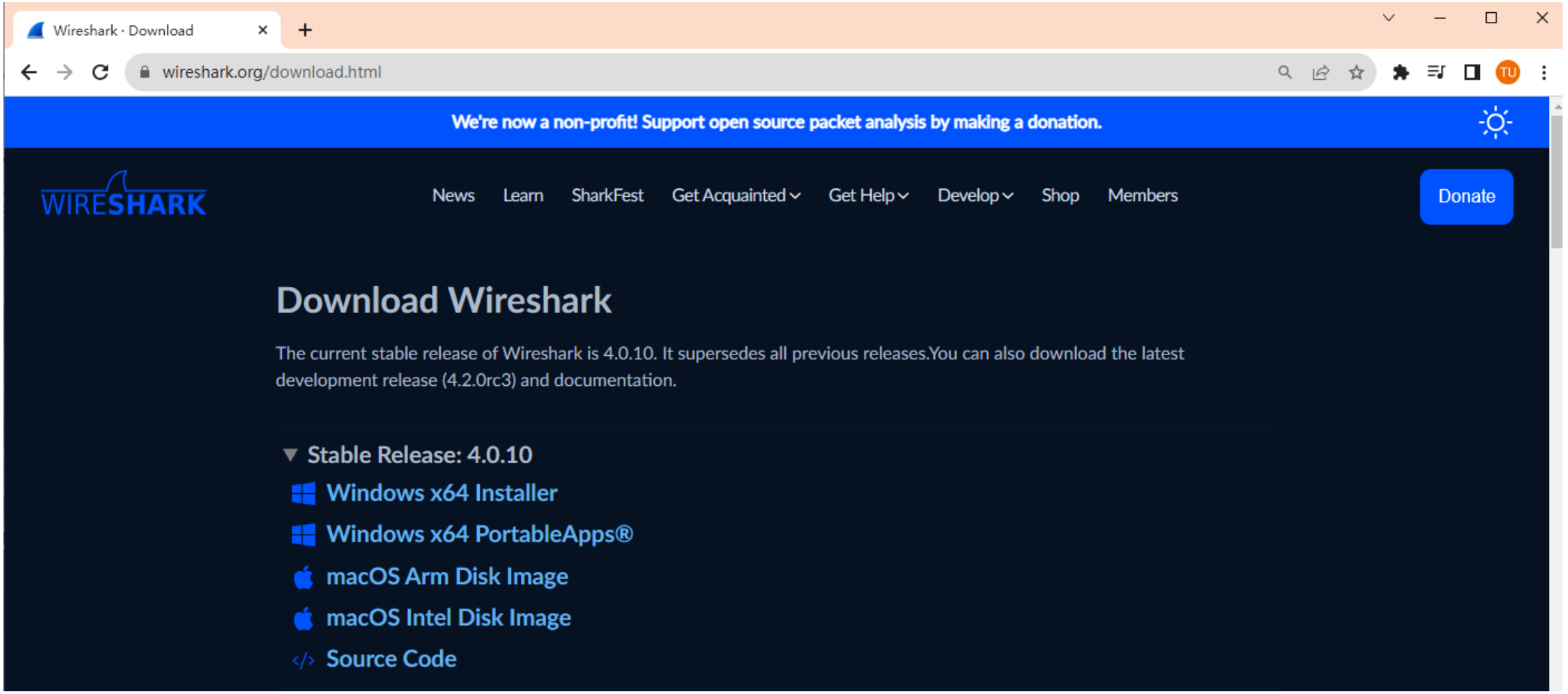
Developers

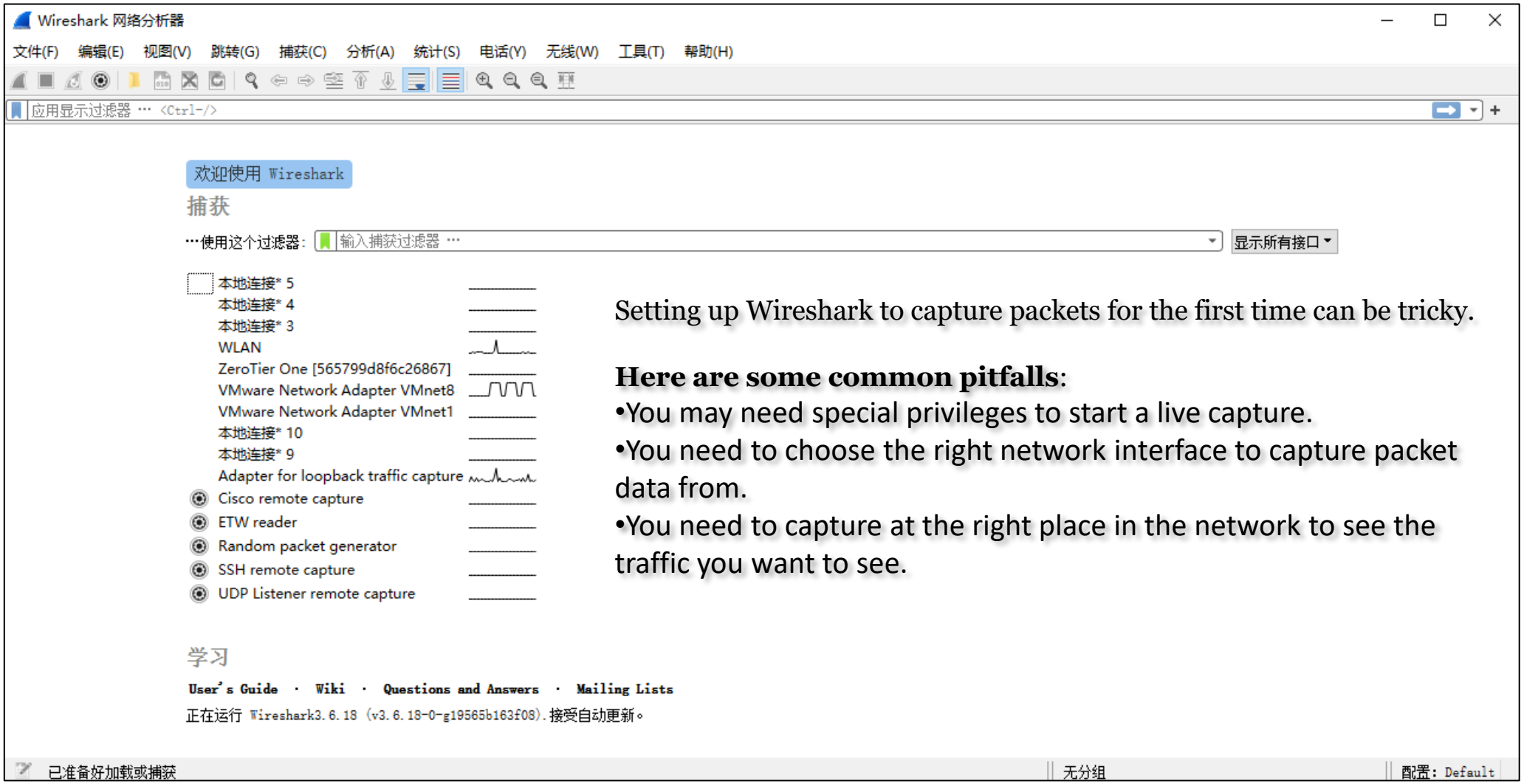


What can wireshark do?

1. **Network troubleshooting**: Wireshark can capture network packets and display detailed packet information, helping to analyze the root cause of network issues.
2. **Network performance optimization**: Wireshark can analyze network traffic patterns, anomalies, and performance problems, enabling network administrators to optimize network performance.
3. **Network security analysis**: Wireshark can assist in detecting and analyzing network attacks, such as denial-of-service attacks or malware propagation, by examining packet contents and patterns.
4. **Protocol development and debugging**: Wireshark supports various protocol decoders and can be used for developing and debugging network protocols by inspecting packet-level details.
5. **Network training and education**: Wireshark is an open-source software that is freely available for users to learn and practice network analysis. It also provides extensive online resources and a supportive community for users to enhance their network analysis skills.

Get Wireshark



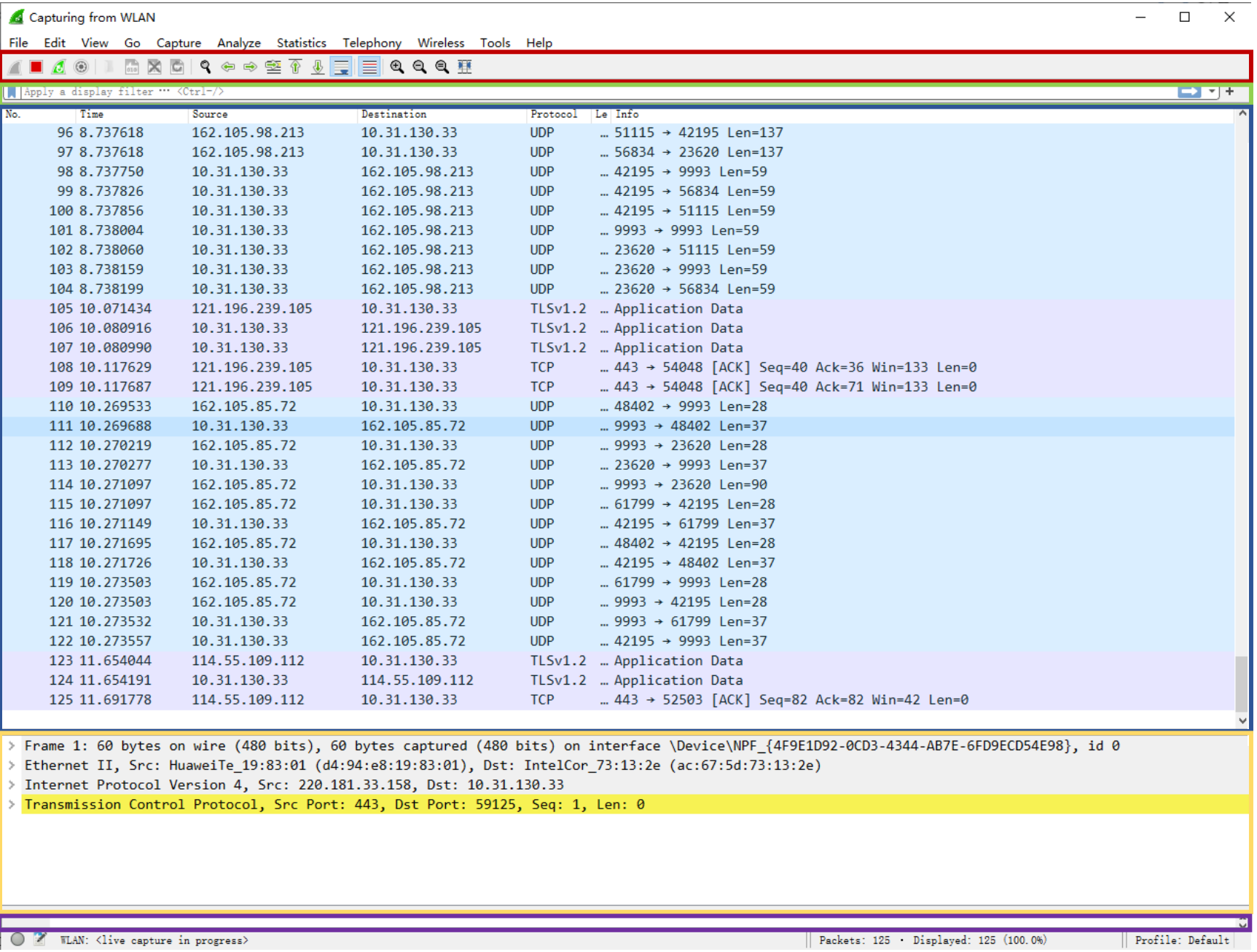


Setting up Wireshark to capture packets for the first time can be tricky.

Here are some common pitfalls:

- You may need special privileges to start a live capture.
- You need to choose the right network interface to capture packet data from.
- You need to capture at the right place in the network to see the traffic you want to see.

An overview of the supported media types can be found at https://gitlab.com/wireshark/wireshark/-/wikis/CaptureSetup/NetworkMedia.



The *main toolbar*

The *filter toolbar*

The *packet list pane*

File

contains items to **open** and merge capture files, **save**, print, or export capture files in whole or in part, and to quit the Wireshark application.

Edit

contains items to **find** a packet, time reference or **mark** one or more packets, handle configuration profiles, and set your preferences; (cut, copy, and paste are not presently implemented).

View

controls the display of the captured data, including **colorization** of packets, zooming the **font**, showing a packet in a separate window, expanding and collapsing trees in packet details, ...

...

The *packet details pane*

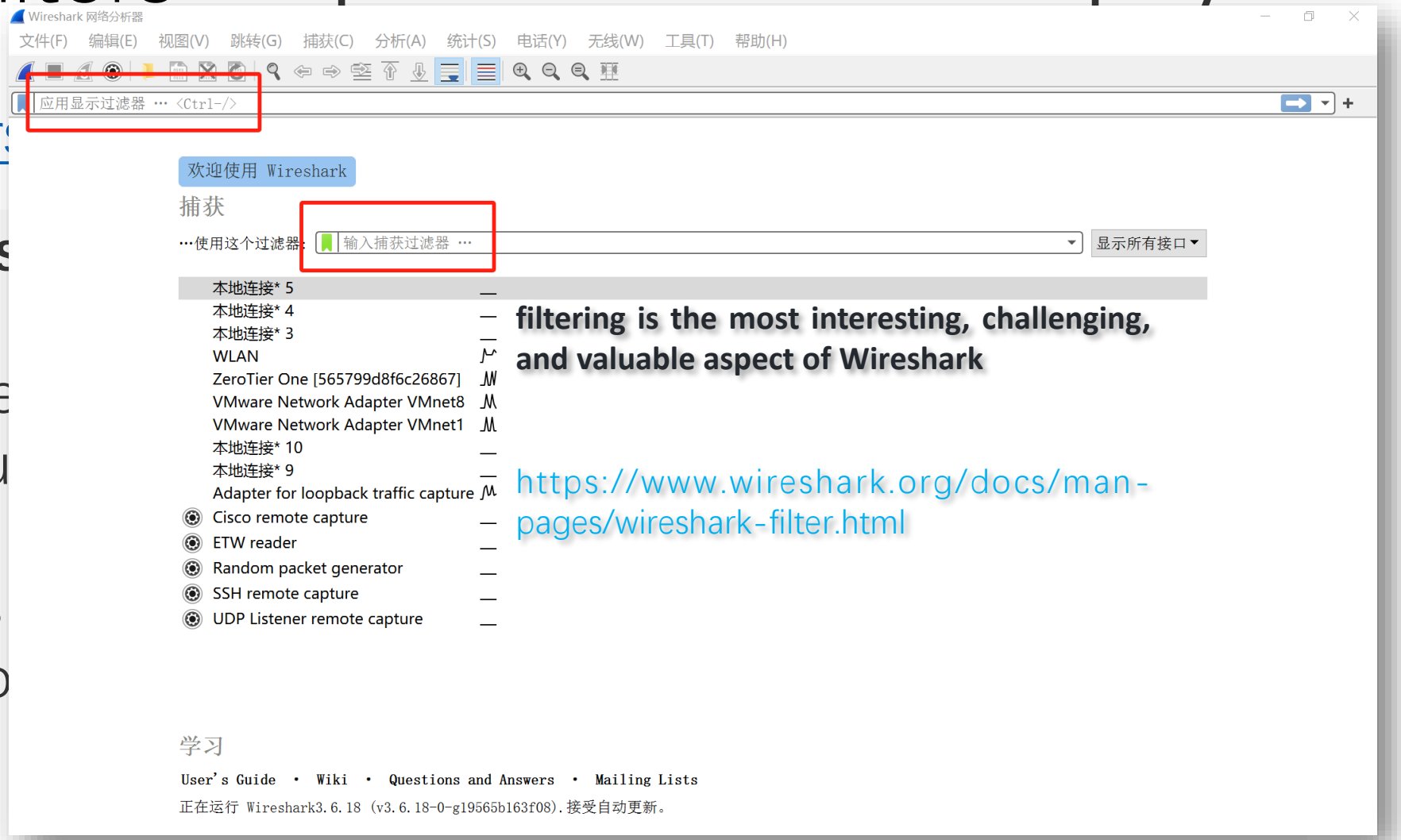
The *packet bytes pane*

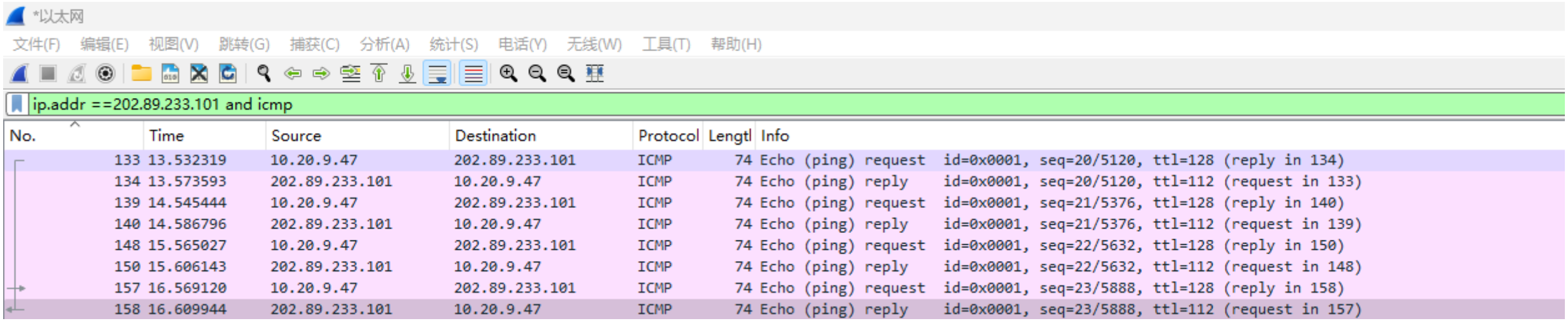
CaptureFilters- Capture filter is not a display filter

- CaptureFilter

Capture filters
display filters
limited and are
The latter are u

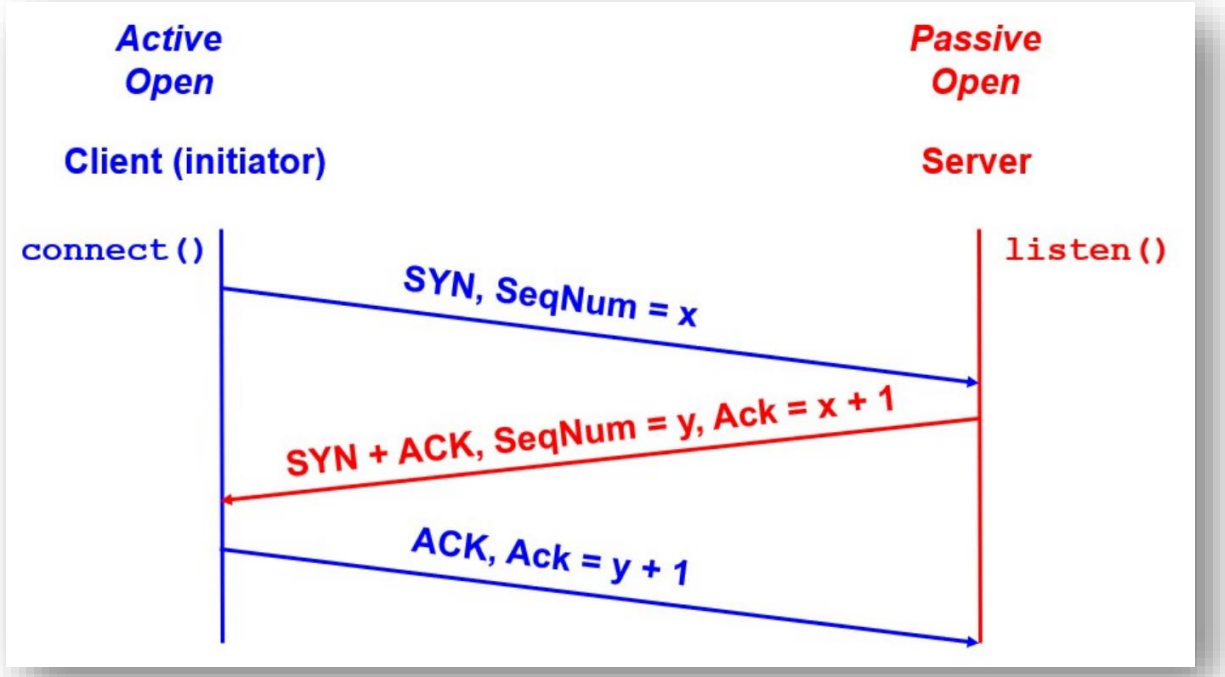
Capture filters
cannot be mo
other hand do
on the fly.




*以太网
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
ip.addr ==202.89.233.101 and icmp
No. Time Source Destination Protocol Lengtl Info
133 13.532319 10.20.9.47 202.89.233.101 ICMP 74 Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 134)
134 13.573593 202.89.233.101 10.20.9.47 ICMP 74 Echo (ping) reply id=0x0001, seq=20/5120, ttl=112 (request in 133)
139 14.545444 10.20.9.47 202.89.233.101 ICMP 74 Echo (ping) request id=0x0001, seq=21/5376, ttl=128 (reply in 140)
140 14.586796 202.89.233.101 10.20.9.47 ICMP 74 Echo (ping) reply id=0x0001, seq=21/5376, ttl=112 (request in 139)
148 15.565027 10.20.9.47 202.89.233.101 ICMP 74 Echo (ping) request id=0x0001, seq=22/5632, ttl=128 (reply in 150)
150 15.606143 202.89.233.101 10.20.9.47 ICMP 74 Echo (ping) reply id=0x0001, seq=22/5632, ttl=112 (request in 148)
157 16.569120 10.20.9.47 202.89.233.101 ICMP 74 Echo (ping) request id=0x0001, seq=23/5888, ttl=128 (reply in 158)
158 16.609944 202.89.233.101 10.20.9.47 ICMP 74 Echo (ping) reply id=0x0001, seq=23/5888, ttl=112 (request in 157)

Another example, if you are only interested packages using the TCP protocol you can just write “TCP” here an then you will only see TCP protocols

10.20.9.47	183.240.166.184	TCP	66 62596 → 36341 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 S
183.240.166.184	10.20.9.47	TCP	66 36341 → 62596 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=12
10.20.9.47	183.240.166.184	TCP	54 62596 → 36341 [ACK] Seq=1 Ack=1 Win=131840 Len=0

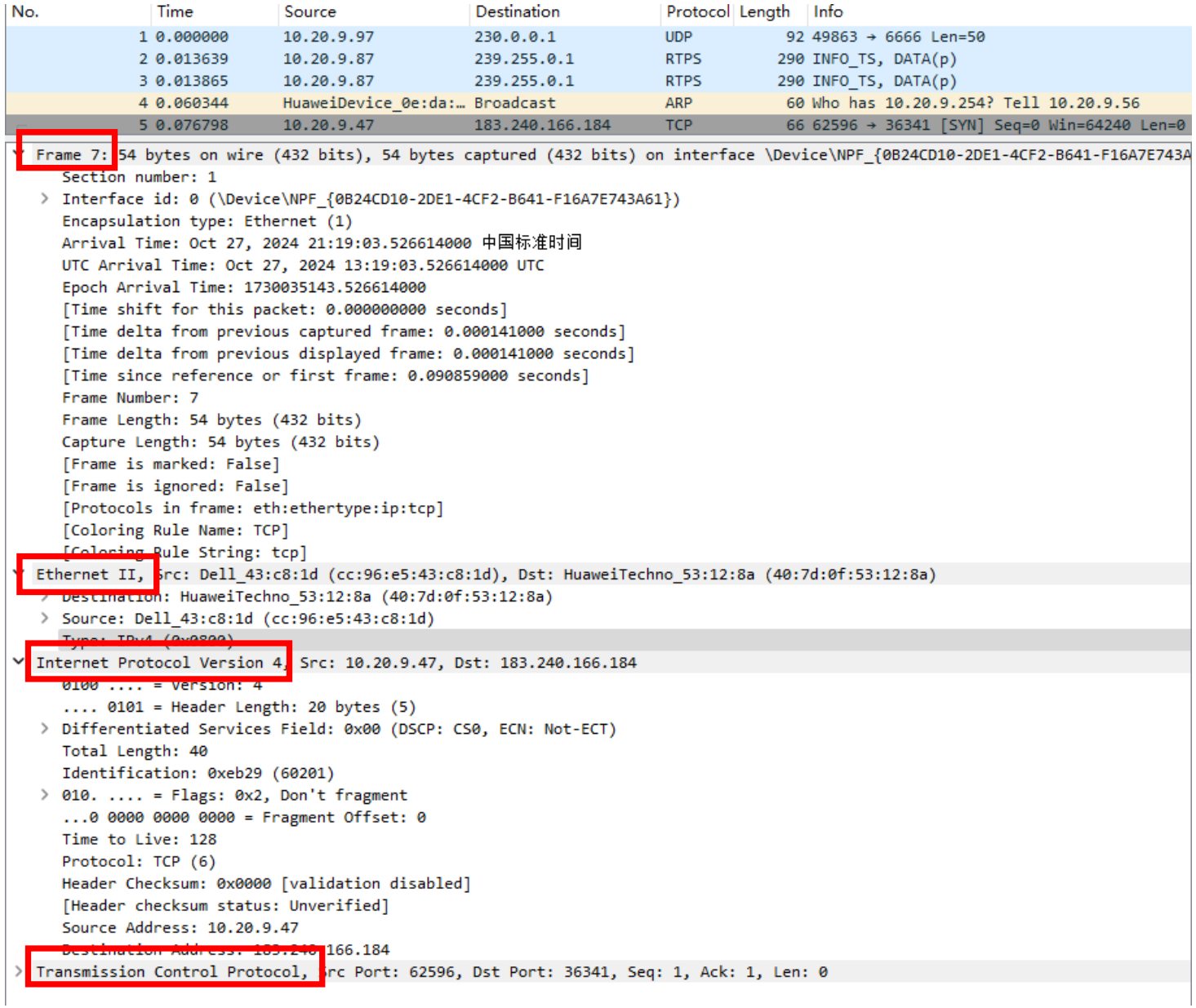


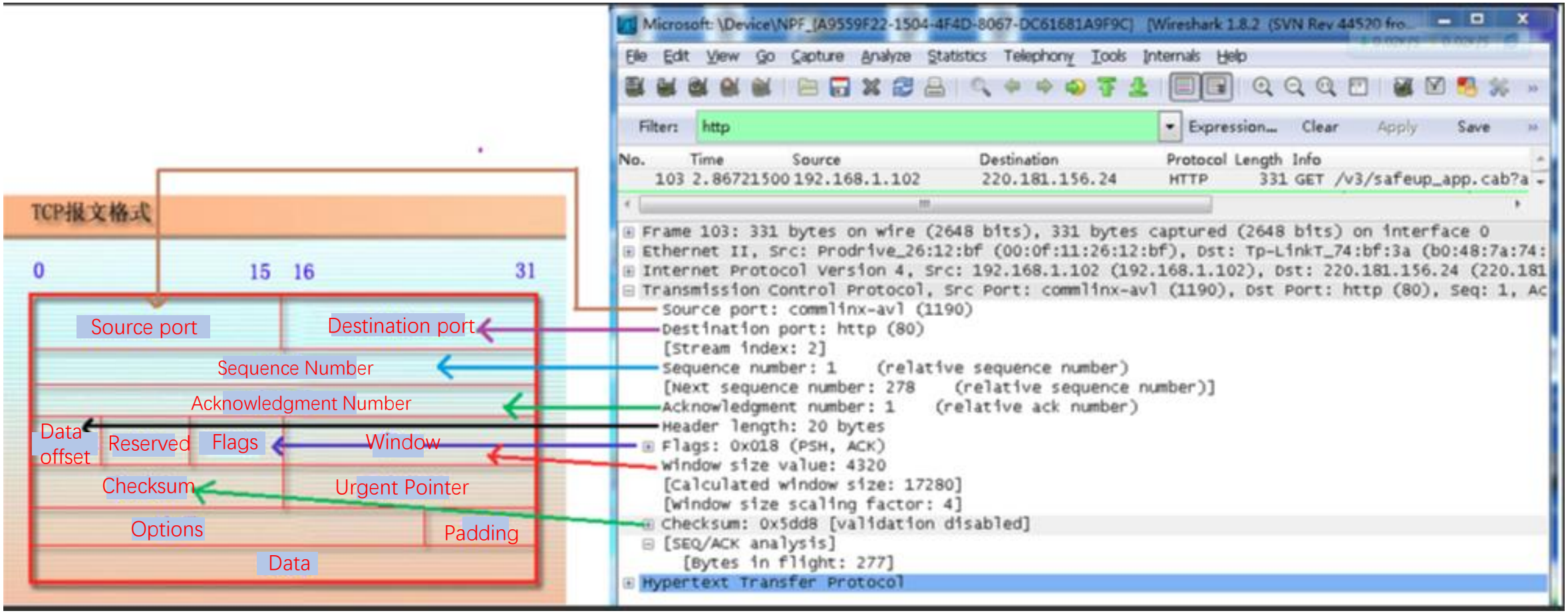
Physical layer

Data Link layer

Network layer

Transport layer




TCP报文格式
0
15
16
31
Source port
Destination port
Sequence Number
Acknowledgment Number
Data offset
Reserved
Flags
Window
Checksum
Urgent Pointer
Options
Padding
Data
Microsoft: \Device\NPF_{A9559F22-1504-4F4D-8067-DC61681A9F9C} [Wireshark 1.8.2 (SVN Rev 44520 fro...
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
103 2.86721500 192.168.1.102 220.181.156.24 HTTP 331 GET /v3/safeup_app.cab?a
Frame 103: 331 bytes on wire (2648 bits), 331 bytes captured (2648 bits) on interface 0
Ethernet II, Src: Prodrive_26:12:bf (00:0f:11:26:12:bf), Dst: Tp-LinkT_74:bf:3a (b0:48:7a:74:
Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 220.181.156.24 (220.181
Transmission Control Protocol, Src Port: commlinx-avl (1190), Dst Port: http (80), Seq: 1, Ac
Source port: commlinx-avl (1190)
Destination port: http (80)
[Stream index: 2]
Sequence number: 1 (relative sequence number)
[Next sequence number: 278 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x018 (PSH, ACK)
Window size value: 4320
[Calculated window size: 17280]
[Window size scaling factor: 4]
Checksum: 0x5dd8 [validation disabled]
[SEQ/ACK analysis]
[Bytes in flight: 277]
Hypertext Transfer Protocol

First handshake packet:

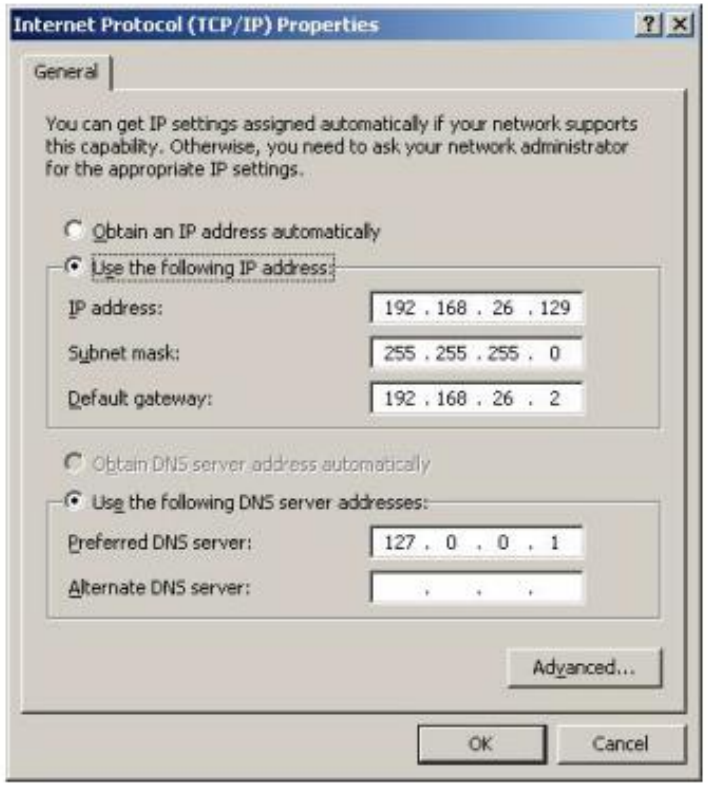
No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily

```
Transmission Control Protocol, Src Port: 14311, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 14311
    Destination Port: 80
    [Stream index: 17]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    > .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ··········S·]
```

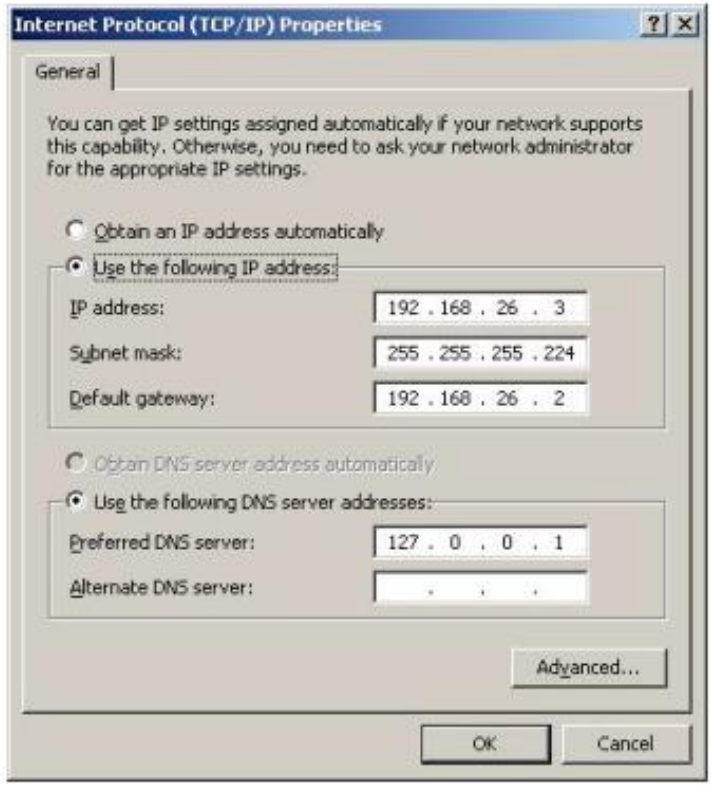
```
0020  02 b7 37 e7 00 50 00 8f  ca 36 00 00 00 00 80 02    ··7··P··  ·6------
```

Two servers, A and B, have the network configuration shown in Figure 1. Server B's subnet mask was supposed to be **255.255.255.0,** but it was accidentally set to **255.255.255.224**. Can they still communicate normally?

Server A

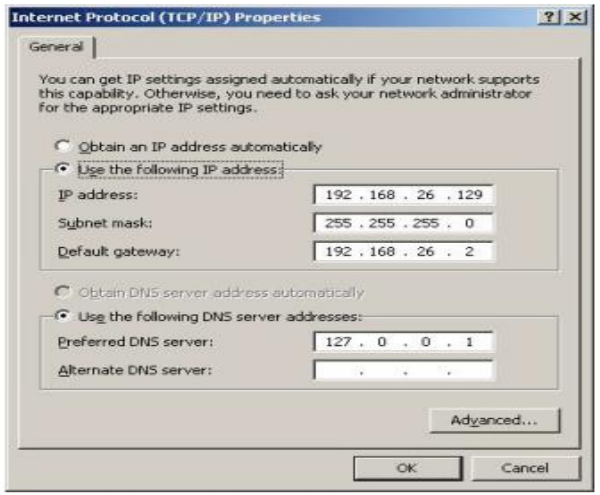


Server B

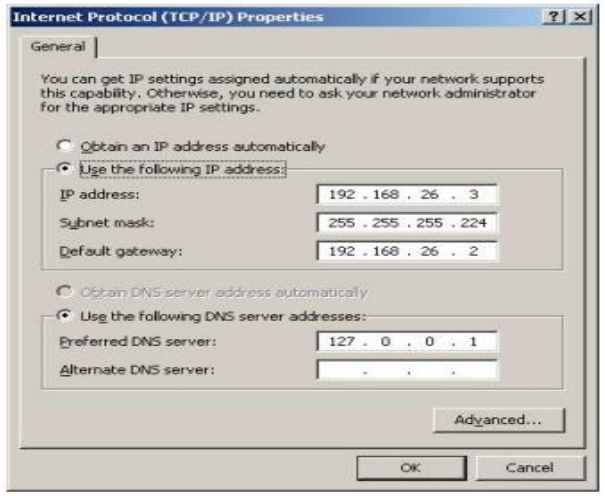


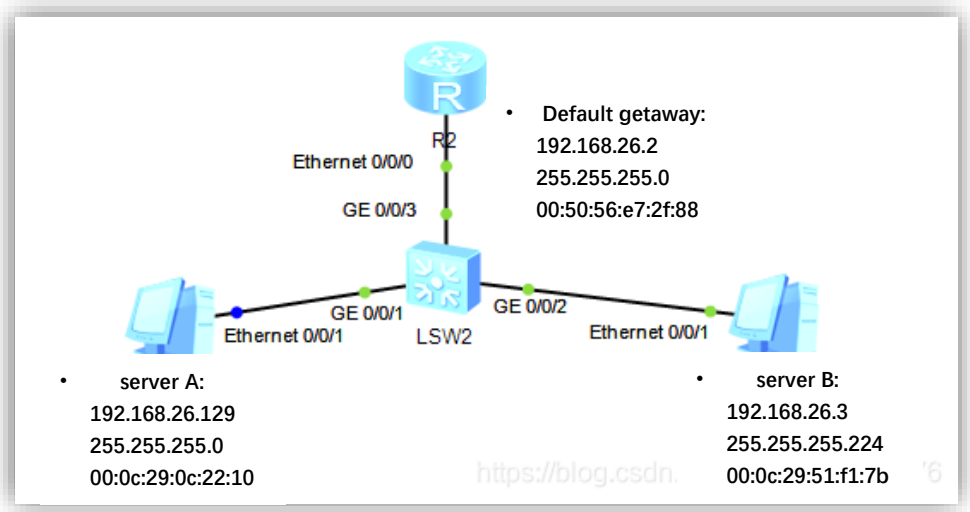
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B



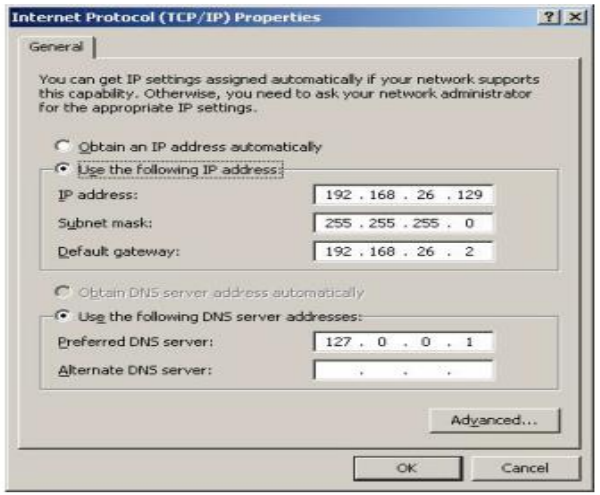


No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

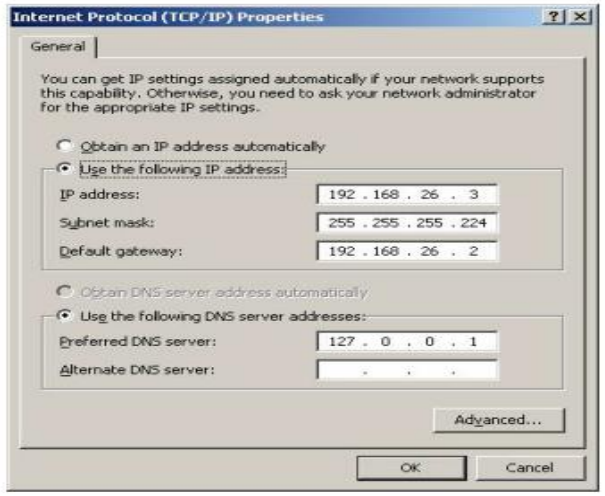
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

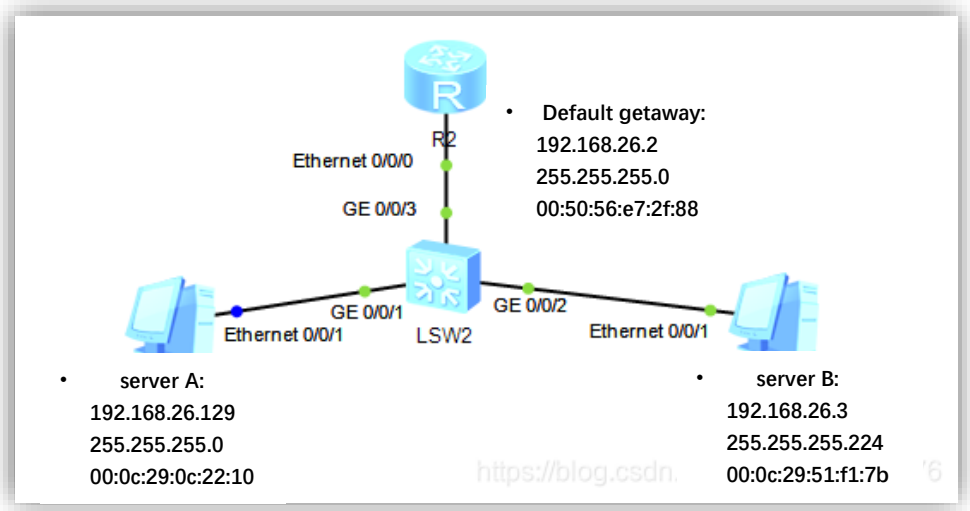
Server B broadcasts an ARP query to request the MAC address of the default gateway, which is 192.168.26.2.

Server A



Server B



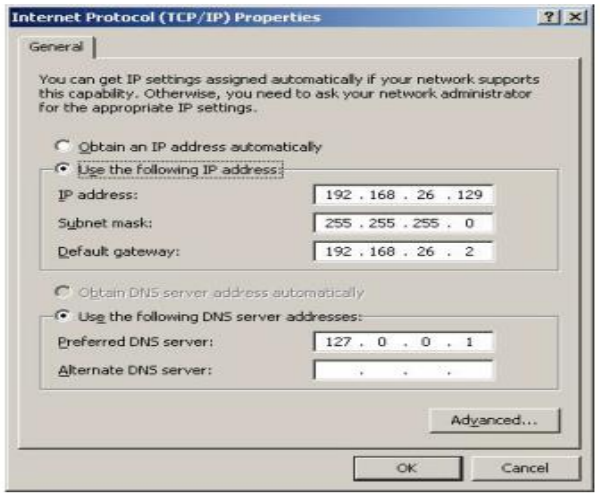


No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

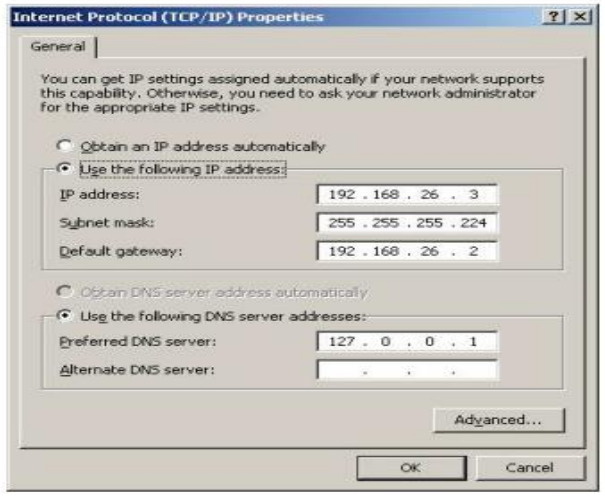
The default gateway 192.168.26.2 replied to B with its own MAC address.

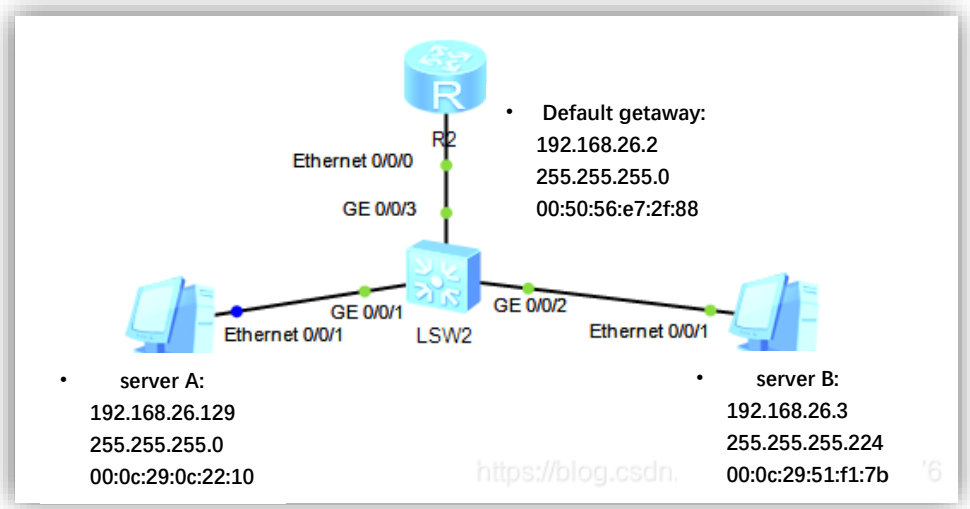
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B





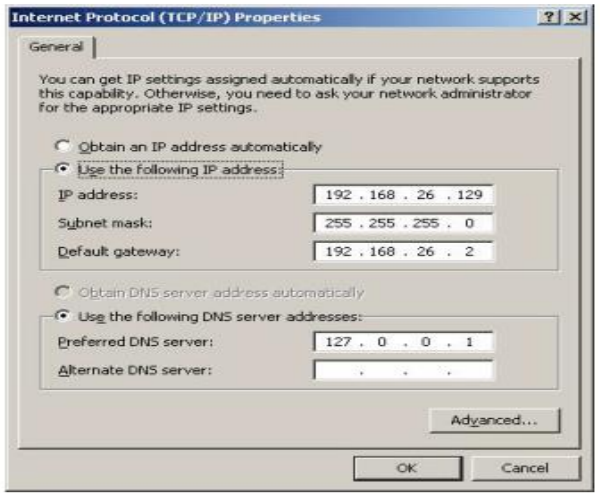
No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

B sends a ping packet with the Destination IP specified as A, which is 192.168.26.129.

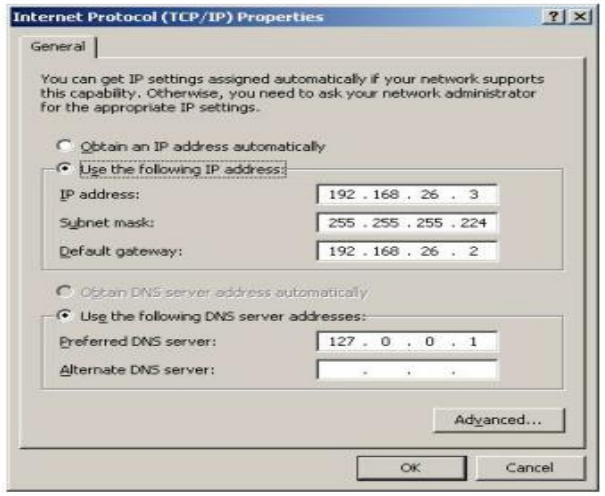
B wants the default gateway to forward the packet to A.

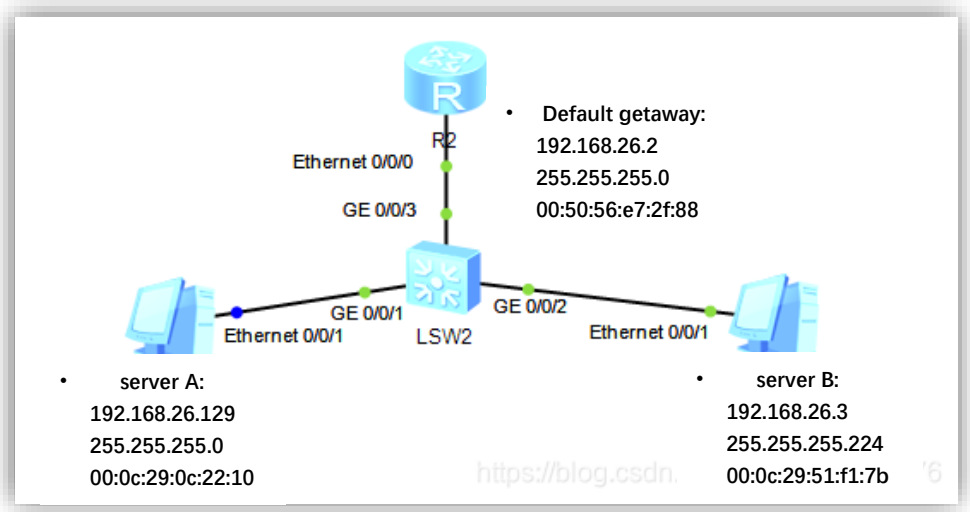
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B



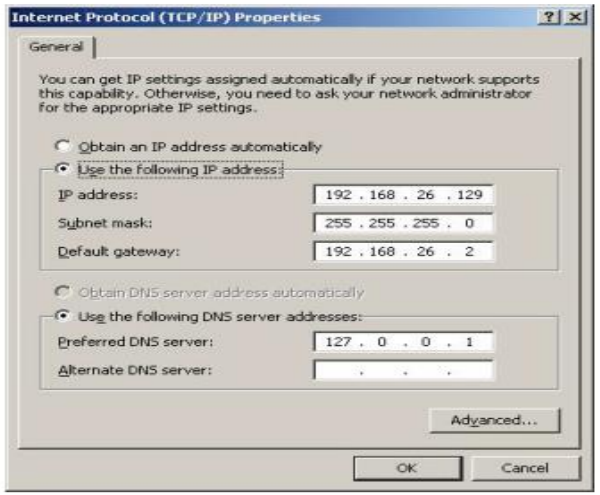


No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

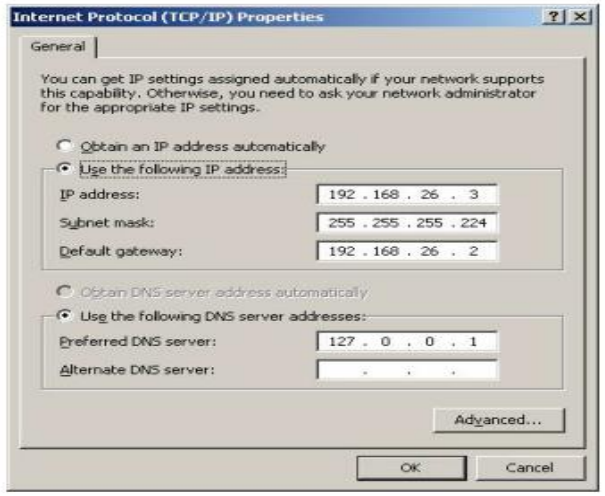
B receives an ARP broadcast sent by A, which queries for B's MAC address.

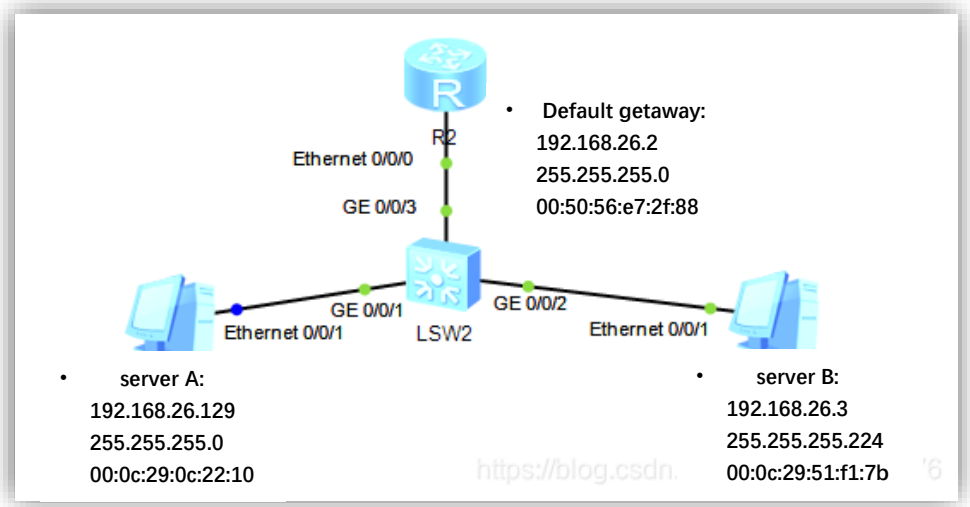
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B



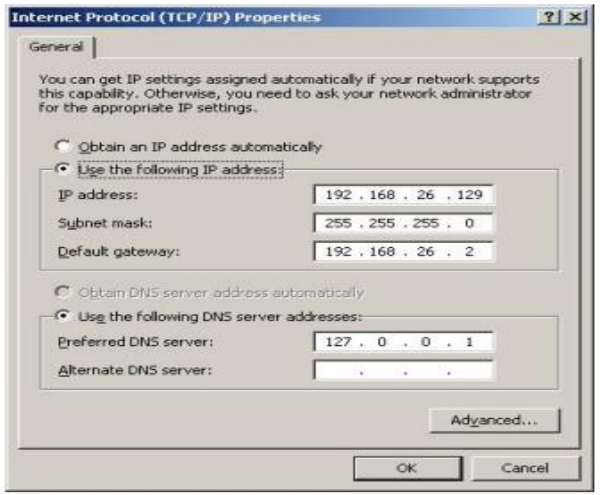


No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

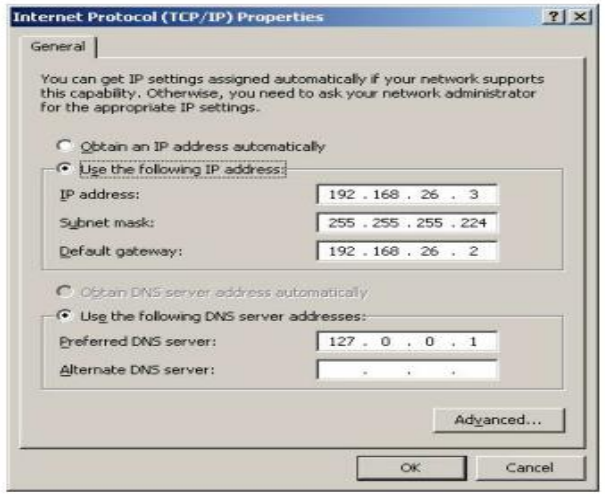
B replies to A's ARP request and informs A of its own MAC address.

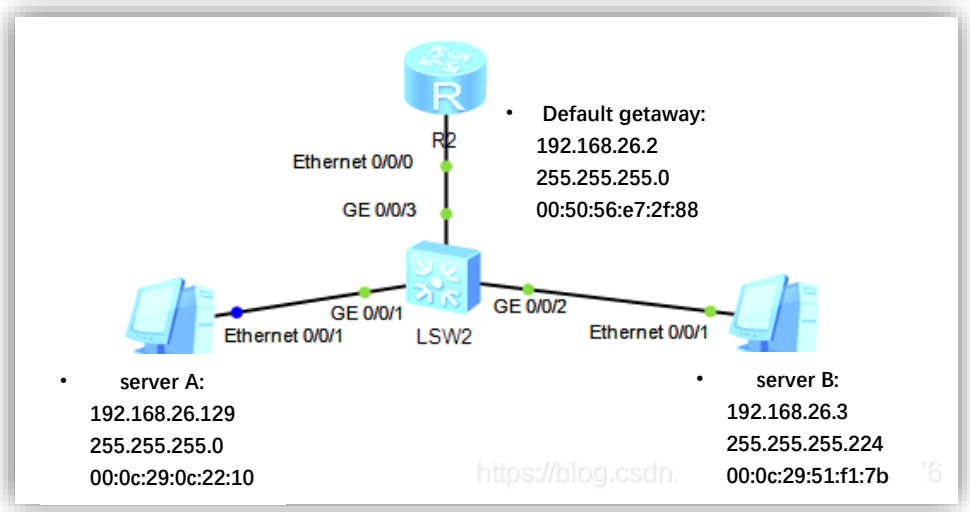
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B



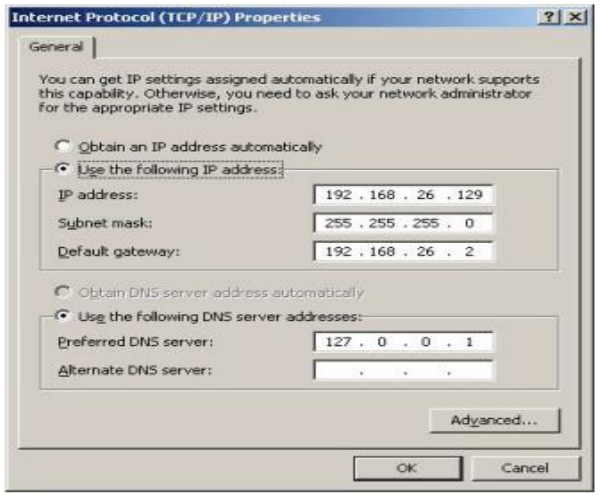


No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

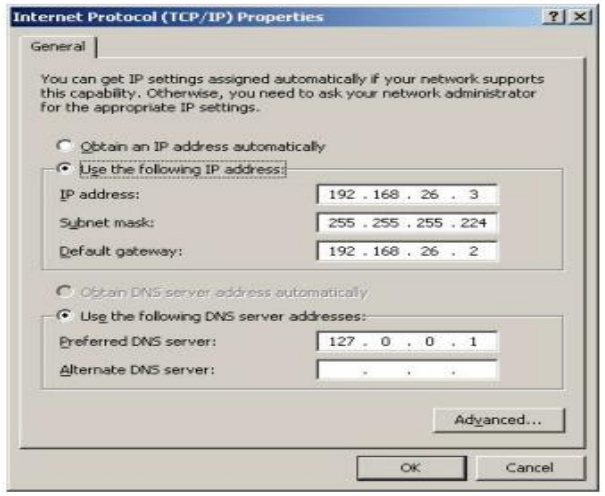
B finally receives the ping reply from A.

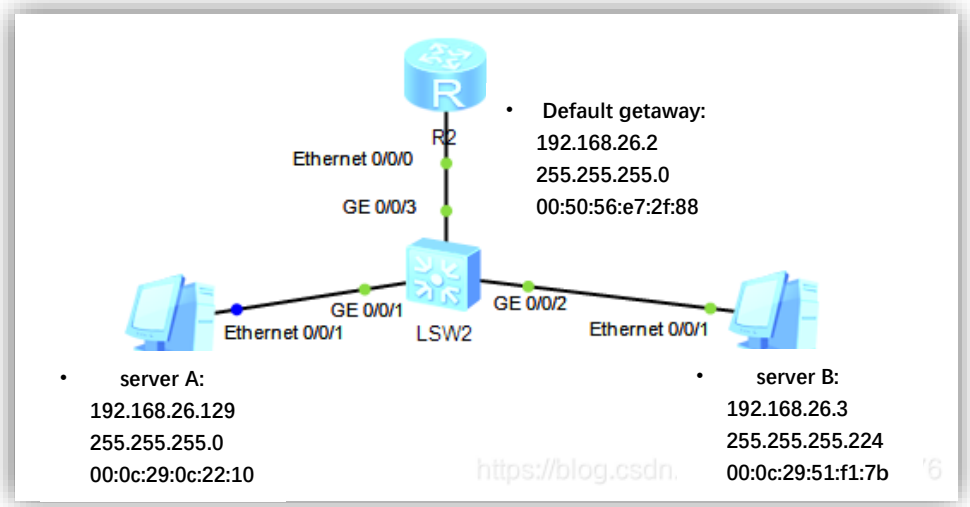
- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

Server A



Server B





No.	Source	Destination	Time	Protocol	Info
1	Vmware_51:f1:7b	Broadcast	2013-04-02 14:18:47.093179	ARP	Who has 192.168.26.2? Tell 192.168.26.3
2	Vmware_e7:2f:88	Vmware_51:f1:7b	2013-04-02 14:18:47.093476	ARP	192.168.26.2 is at 00:50:56:e7:2f:88
3	192.168.26.3	192.168.26.129	2013-04-02 14:18:47.093500	ICMP	Echo (ping) request id=0x0200, seq=4352/17, ttl=128
4	Vmware_0c:22:10	Broadcast	2013-04-02 14:18:47.094076	ARP	Who has 192.168.26.3? Tell 192.168.26.129
5	Vmware_51:f1:7b	Vmware_0c:22:10	2013-04-02 14:18:47.094104	ARP	192.168.26.3 is at 00:0c:29:51:f1:7b
6	192.168.26.129	192.168.26.3	2013-04-02 14:18:47.094393	ICMP	Echo (ping) reply id=0x0200, seq=4352/17, ttl=128
7	192.168.26.3	192.168.26.129	2013-04-02 14:18:48.084739	ICMP	Echo (ping) request id=0x0200, seq=4608/18, ttl=128
8	192.168.26.129	192.168.26.3	2013-04-02 14:18:48.085416	ICMP	Echo (ping) reply id=0x0200, seq=4608/18, ttl=128
9	192.168.26.3	192.168.26.129	2013-04-02 14:18:49.098809	ICMP	Echo (ping) request id=0x0200, seq=4864/19, ttl=128
10	192.168.26.129	192.168.26.3	2013-04-02 14:18:49.099351	ICMP	Echo (ping) reply id=0x0200, seq=4864/19, ttl=128

Repeated ping requests and ping replies.

- Server A: 00:0c:29:0c:22:10
- Server B: 00:0c:29:51:f1:7b
- Default Gateway: 00:50:56:e7:2f:88

NEXT

- More application examples of Wireshark
- Review some theoretical concepts
- ASSIGNMENT 3 - related