

A Short Intro to Wireshark

- WireShark is a packet analyzer
 - also packer sniffer
- What does Wireshark do?
 - it captures packets on the NIC (Network Interface Card)
 - analyzes the protocol headers
 - filters packets according to your preferences
 - displays packets with time stamps, headers, content and helpful information
 - protocol, flows, timing, etc.
 - provides tool to analyzes your traffic

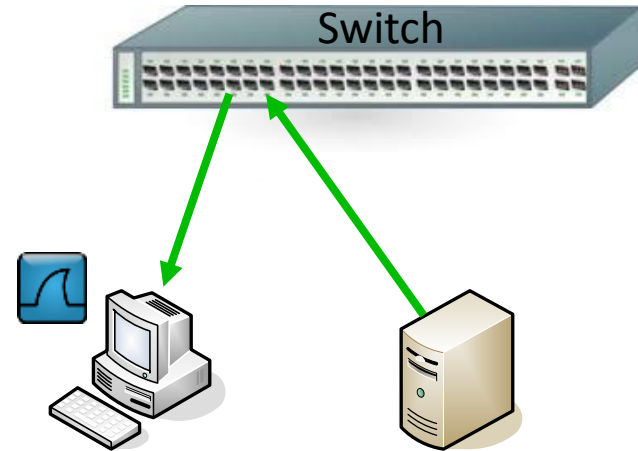
What is Wireshark good for?

- Every Internet user
 - investigate with whom your computer/your programs/your browser is communicating
 - understand your network performance and investigate potential performance problems
 - Why is the connection to a server so slow?
- Developer of networking applications
 - debug your code by analyzing the exchanged packets
 - check for network-originated performance problems
- Network administrator / network operator
 - check for network problems
 - identify misbehaving computers and attacks from within or outside the network

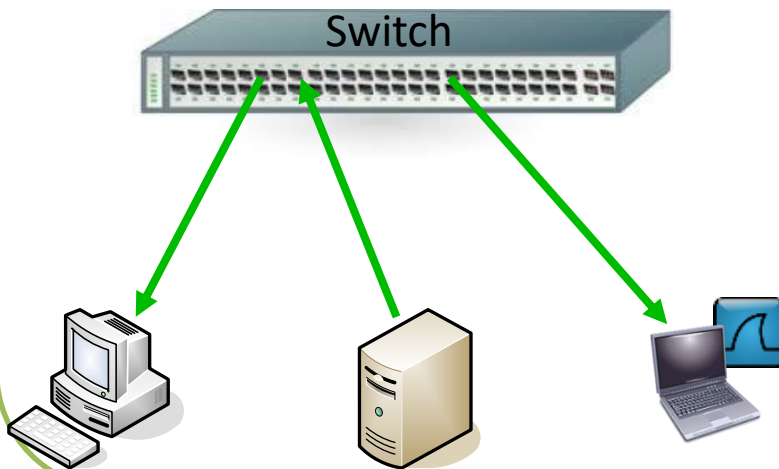
Collecting Packets

On the client:

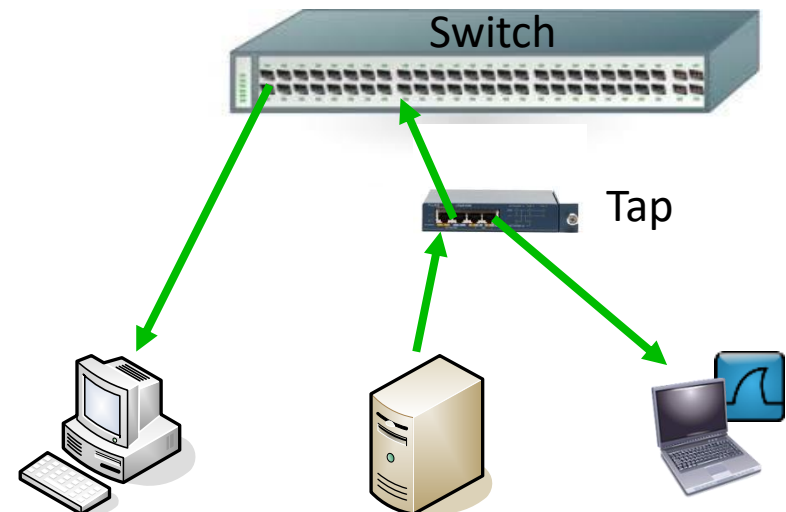
- only sees packets to/from the client's address and broadcast traffic
- cannot sniff other traffic in the network



Span/Mirror: Switch copies selected ports, hosts, vlans, or traffic patterns to a monitor port



Tap: Device that forwards traffic and puts a copy on another port



What can you analyze with Wireshark

- Standard usage:
 - live traffic on Ethernet or WiFi NIC
 - you may deploy capture filter to reduce the number of packets stored by WireShark
- Analyze traffic dumped in a file:
 - you can dump traffic into a file using command line applications like tcpdump or rawcap that run in the background and analyze the file later
 - you can use rawcap to dump computer internal (localhost) traffic to a file and, rawcap needs admin rights
- Other types of traffic
 - Wireshark also supports PPP (DSL), DOCSIS (cable modem), USB, or Bluetooth traffic

And now try it ...

The screenshot shows the Wireshark 1.12.4 interface with the filter 'arp' applied. Two packets are visible in the packet list:

No.	Time	Source	Src Port	Destination	Dst Port	Length	Payload	Protocol	Info
277	16.63325	Hewlett_8a:f6:		Broadcast		60		ARP	who has 141.37.29.20? Tell 141.37.29.100
368	19.62301	Hewlett_8a:f6:		Broadcast		60		ARP	who has 141.37.29.22? Tell 141.37.29.100

The packet details pane for packet 368 shows the following structure:

- Frame 368: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: Hewlett_8a:f6:27 (00:21:5a:8a:f6:27), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (1)
 - Sender MAC address: Hewlett_8a:f6:27 (00:21:5a:8a:f6:27)
 - Sender IP address: 141.37.29.100 (141.37.29.100)
 - Target MAC address: 03:05:01:02:01:04 (03:05:01:02:01:04)
 - Target IP address: 141.37.29.22 (141.37.29.22)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

Offset	Hex	ASCII
0000	ff ff ff ff ff ff 00 21 5a 8a f6 27 08 06 00 01!Z..'.d
0010	08 00 06 04 00 01 00 21 5a 8a f6 27 8d 25 1d 64!Z..%.d
0020	03 05 01 02 01 04 8d 25 1d 16 00 00 00 00 00 00%.....
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

The status bar at the bottom indicates: Frame (frame), 60 bytes | Packets: 464 · Displayed: 2 (0,4%) · Dropped: 0 (0,0%) | Profile: Default