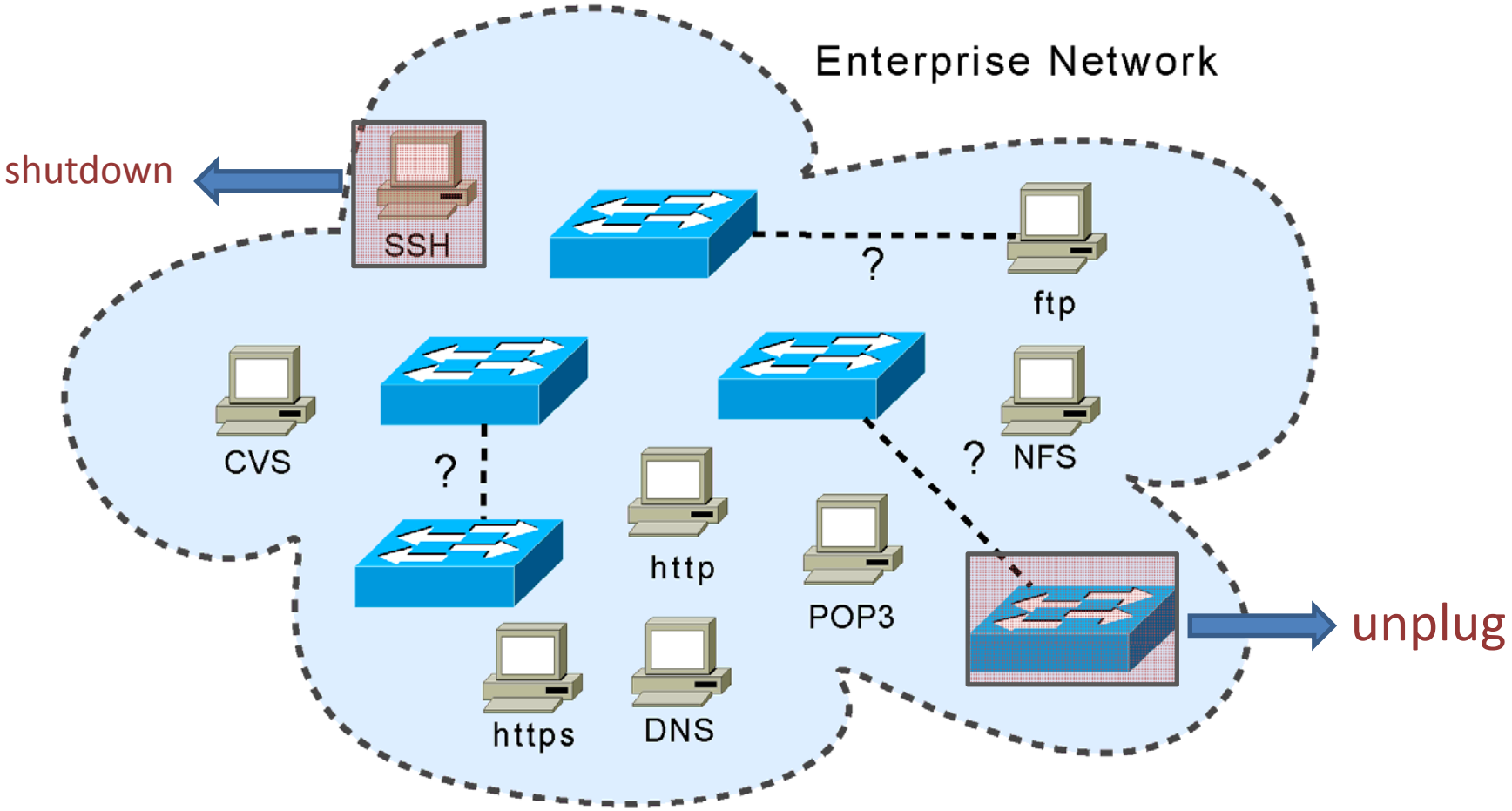
Automated Service Discovery for Enterprise Network Management

Stony Brook University

March 4, 2009

Motivation



what happen when a network device is ***unplugged*** for maintenance?

The Question

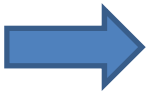
Given an enterprise network, how to:

- Show how switches, routers, hosts and servers are physically connected? -> Topology
- Automatically discover all network services -> Service Discovery
- Find out which hosts access to which services -> Host Dependency

Contribution

- A real-time visibility network management tool that shows
  1. map of enterprise network services
  2. map of hosts associated with a service
  3. Layer2 and Layer3 topology

Talk outline



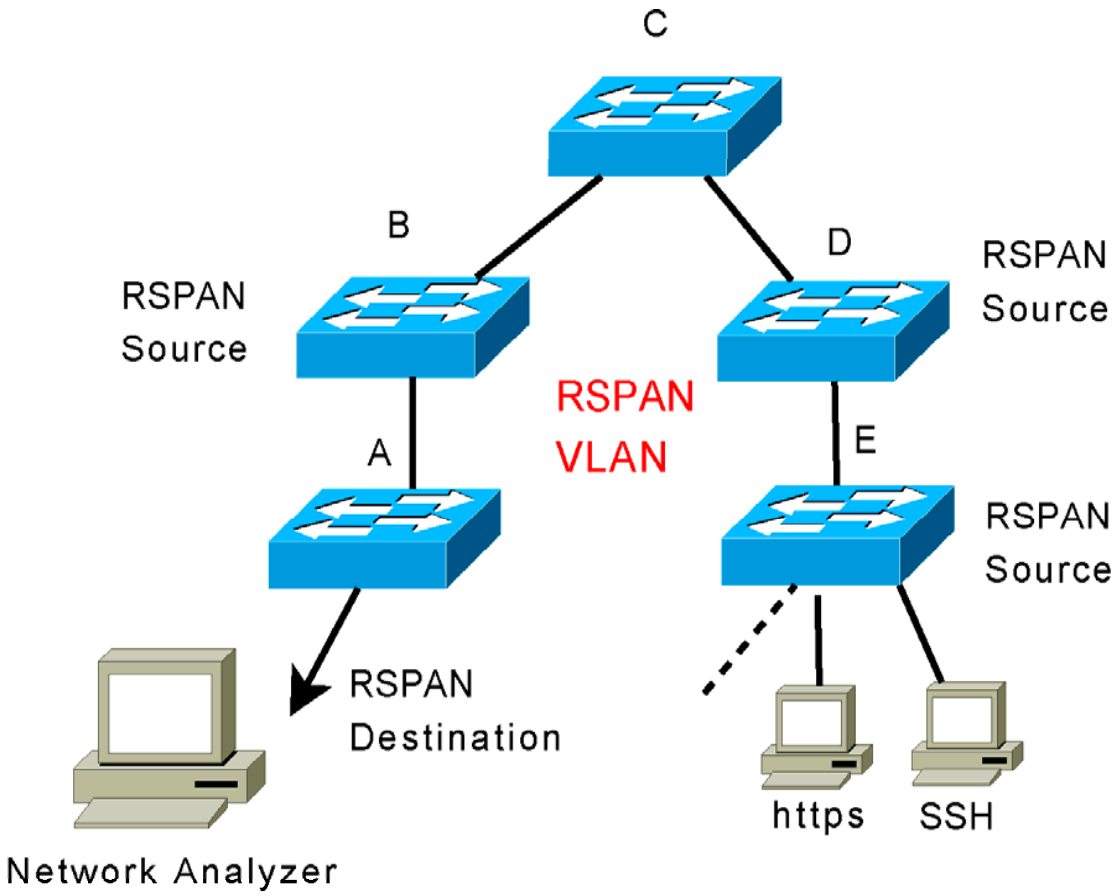
- Service Discovery
- System Design
- Network Topology Discovery
- Experiments
- Conclusion

Service Discovery: Basic idea

- Audit the network by monitoring traffic
  - Port mirroring -> comprehensive?
  - Solution: Weighted Round-robin RSPAN
- Recognize services by
  - Port Number -> reliable?
  - Payload -> too expensive, performance?
  - Solution: Integrated active and passive Service Discovery tool

Traffic Monitor: RSPAN

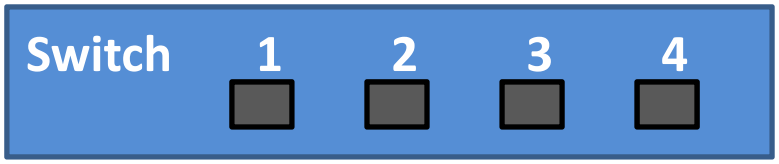
- RSPAN: Remote Switched Port Analyzer



- Switch set as RSPAN source will copy traffic into RSAPN VLAN and forwarded to RSPAN destination port
- We can monitor all traffic at one port!?
- Timing? Packet loss? Performance?

Weighted Round-robin RSPAN

- To reduce RSPAN overhead, we plan to
  - Query the traffic load of each ports on a switch
  - Change the RSPAN interval on source port based on its load
  - Balance the RSPAN traffic on each ports

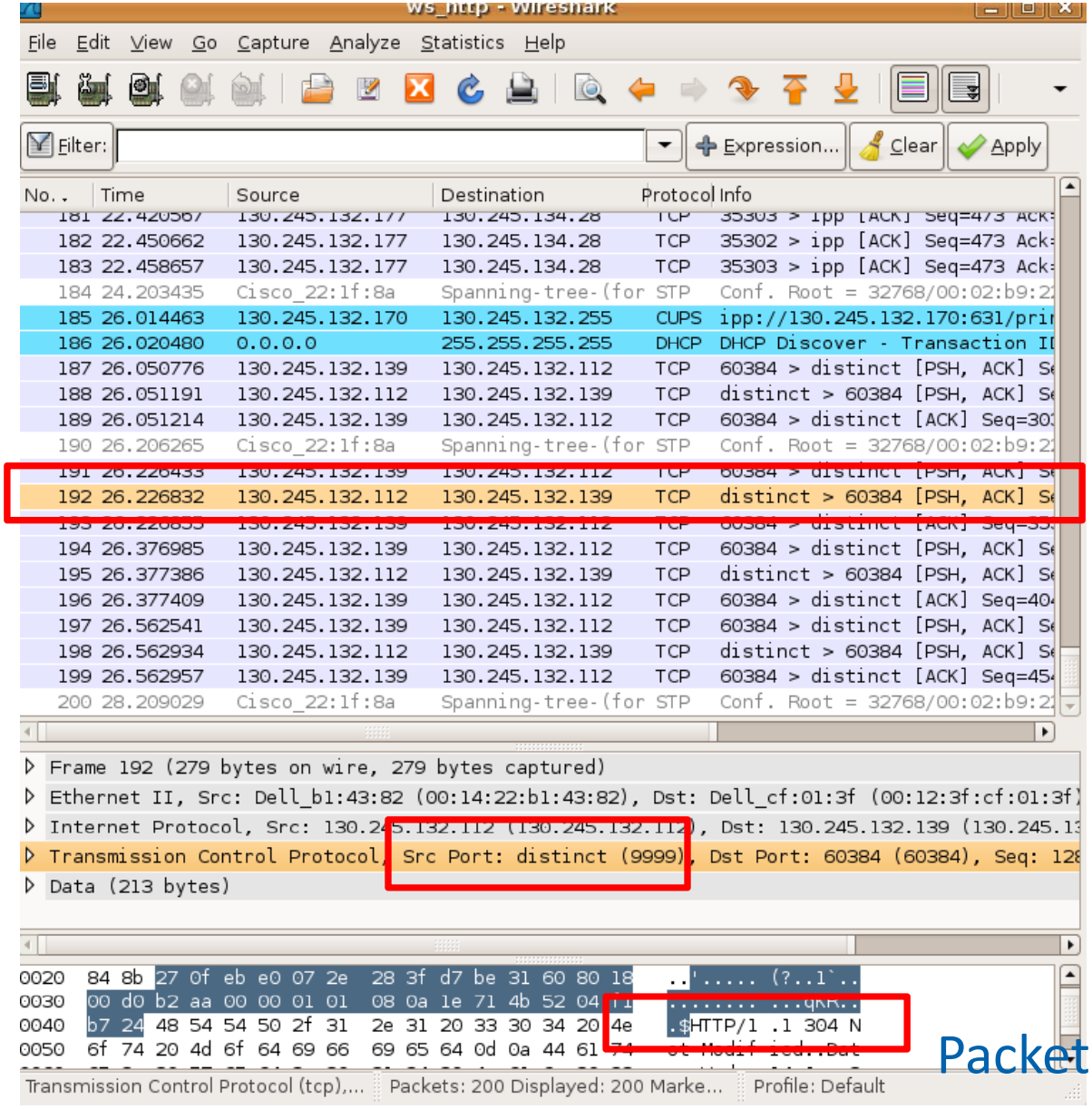


Port	Traffic (Mbit/sec)	Interval (sec)
1	5	20
2	10	15
3	15	10
4	20	5

Service Discovery: Wireshark

- Wireshark (Ethereal)
  - Network protocol analyzer
  - Capture online / offline network data
  - Inspect hundreds of protocols
- Typical uses
  - Know application behavior on the wire
  - Monitor network
  - Identify application dependency

Wireshark Screenshot



http connection running on port 9999

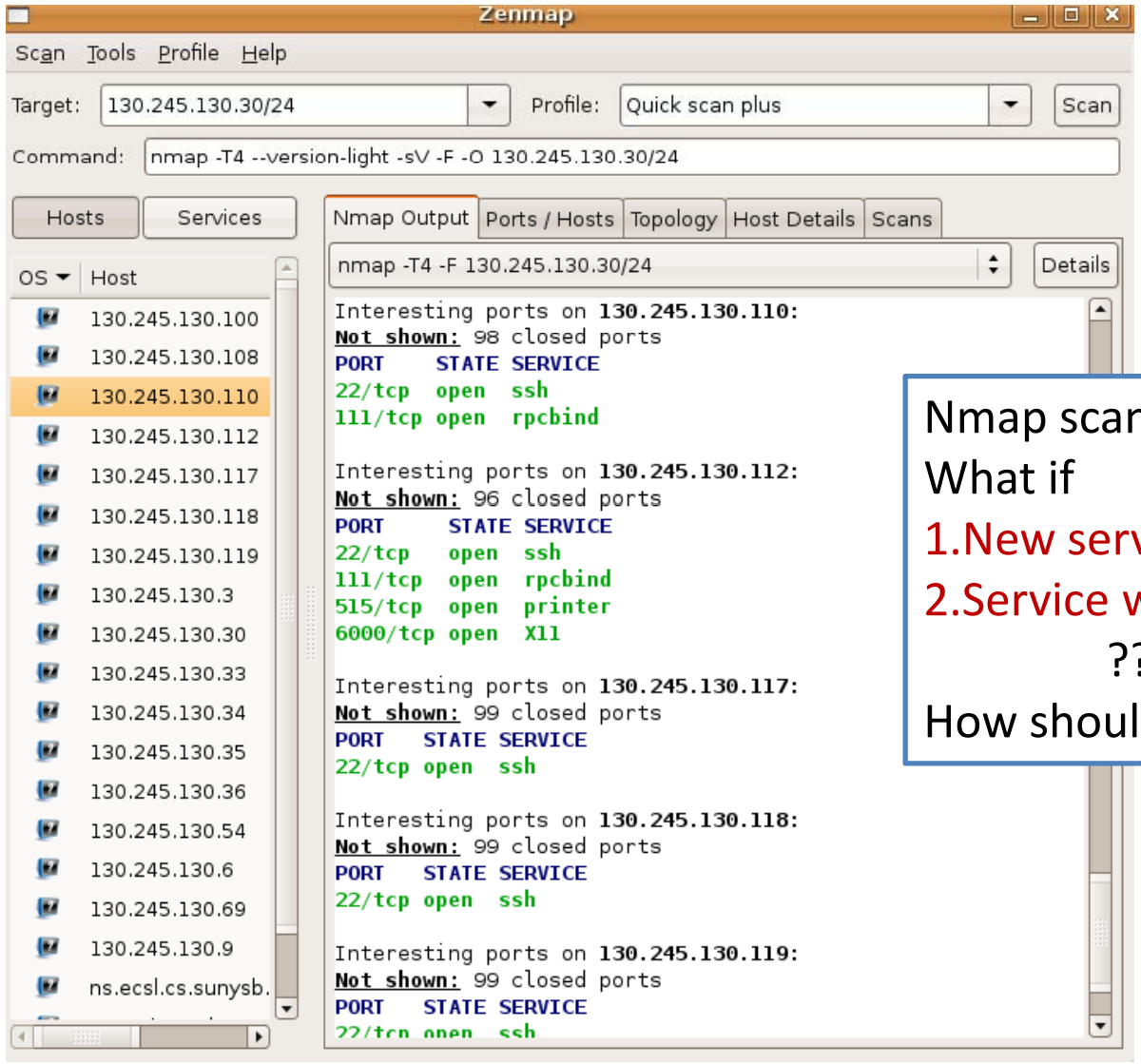
Wireshark can not detect it's a http protocol!

Packet content

Service Discovery: Nmap

- Nmap
  - A tool for host discovery, port scanning, version detection, and OS detection.
- Typical uses
  - Auditing the security of a computer
  - Identifying open ports
  - Monitoring host or service uptime
  - Scan large networks

Nmap GUI: Zenmap



Nmap scan result…

What if

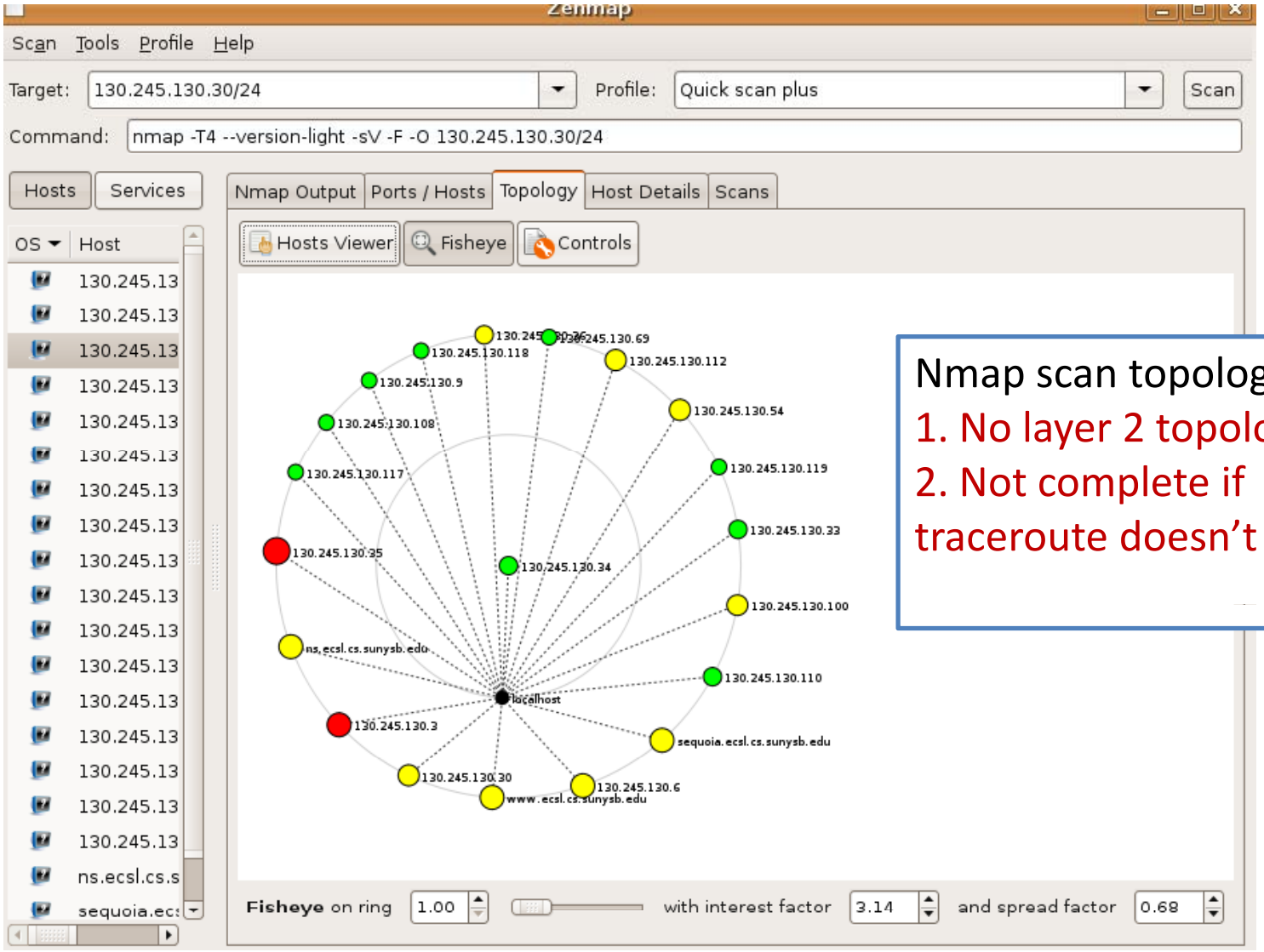
1.New service coming up

2.Service was shutdown

??

How should we scan?

Nmap GUI: Zenmap



Nmap scan topology…

1. No layer 2 topology
2. Not complete if traceroute doesn't work

Service Discovery: Comparison

	Wireshark (Ethereal)	Nmap
Monitor / Scan Network	Yes / No	No / Yes
# Supported Protocols	800~850	Around 700
Detect Host Dependency	Yes	No
Service Detection	Poor	Good

Service Discovery: Solution

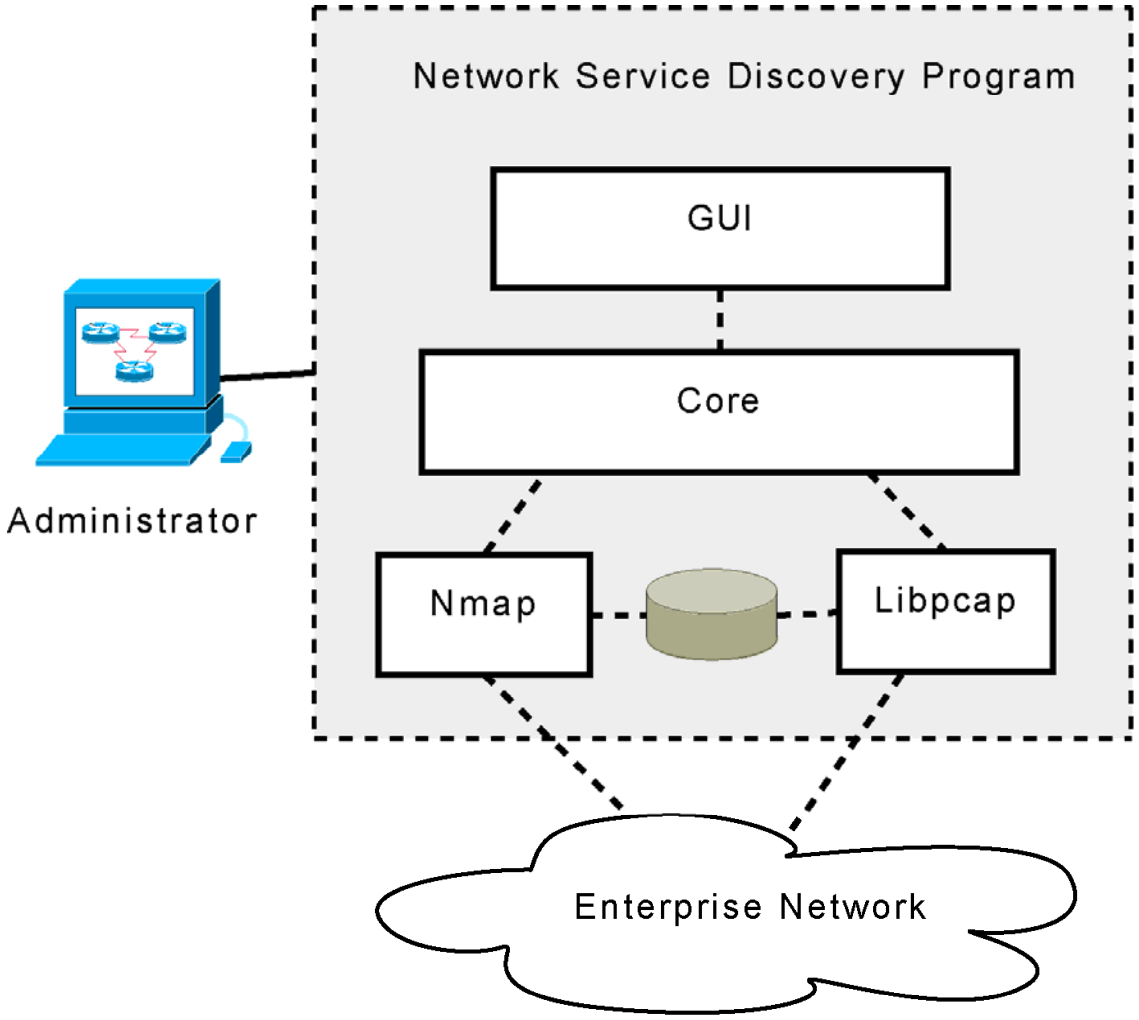
A tool that can

- Efficiently monitor the entire network and find dependency like Wireshark
- Actively probing a host to find out its open services like Nmap
- Provide Layer 2 and Layer 3 network topology

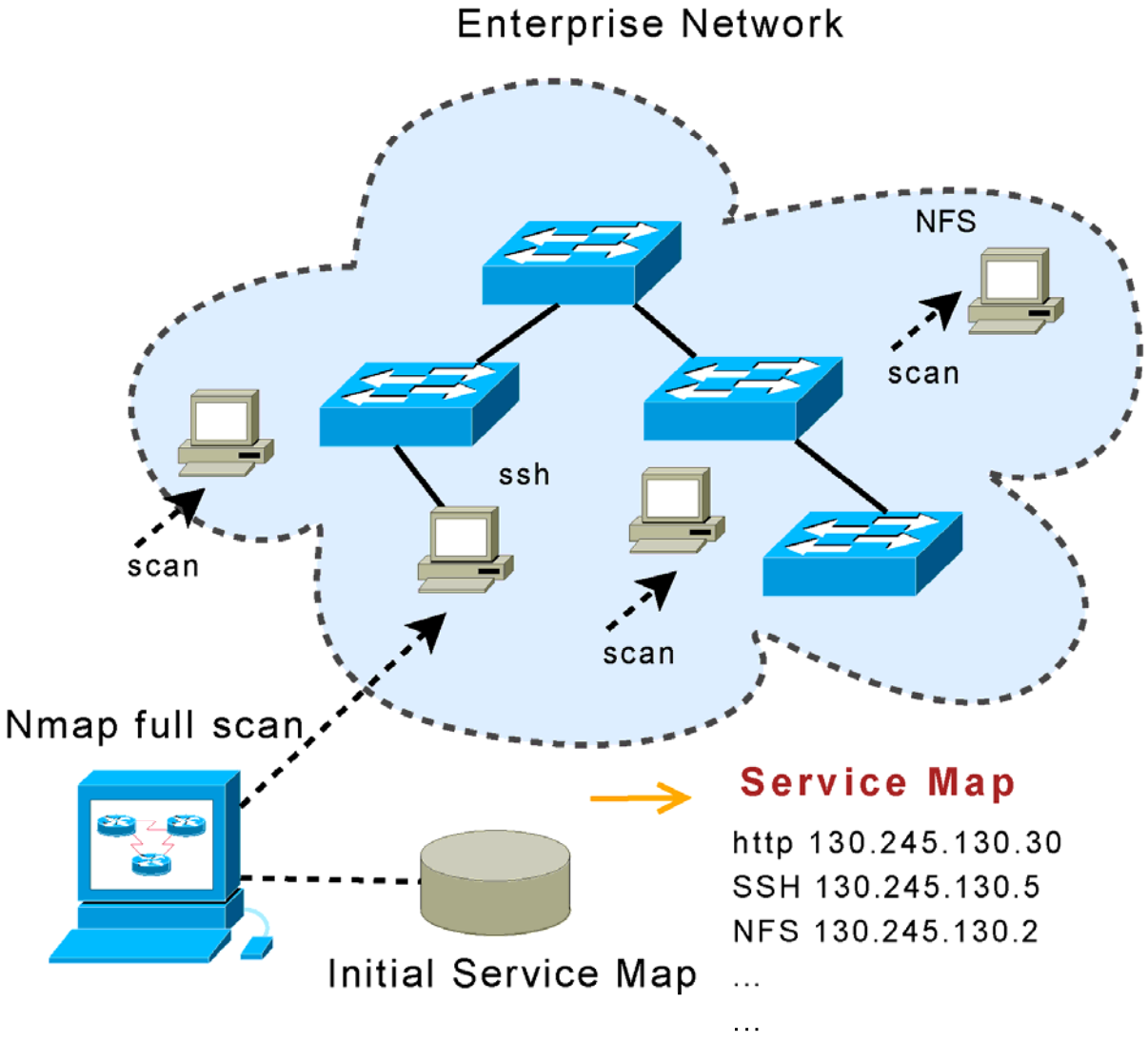
Talk outline

- Service Discovery
- System Design
- Network Topology Discovery
- Experiments
- Conclusion

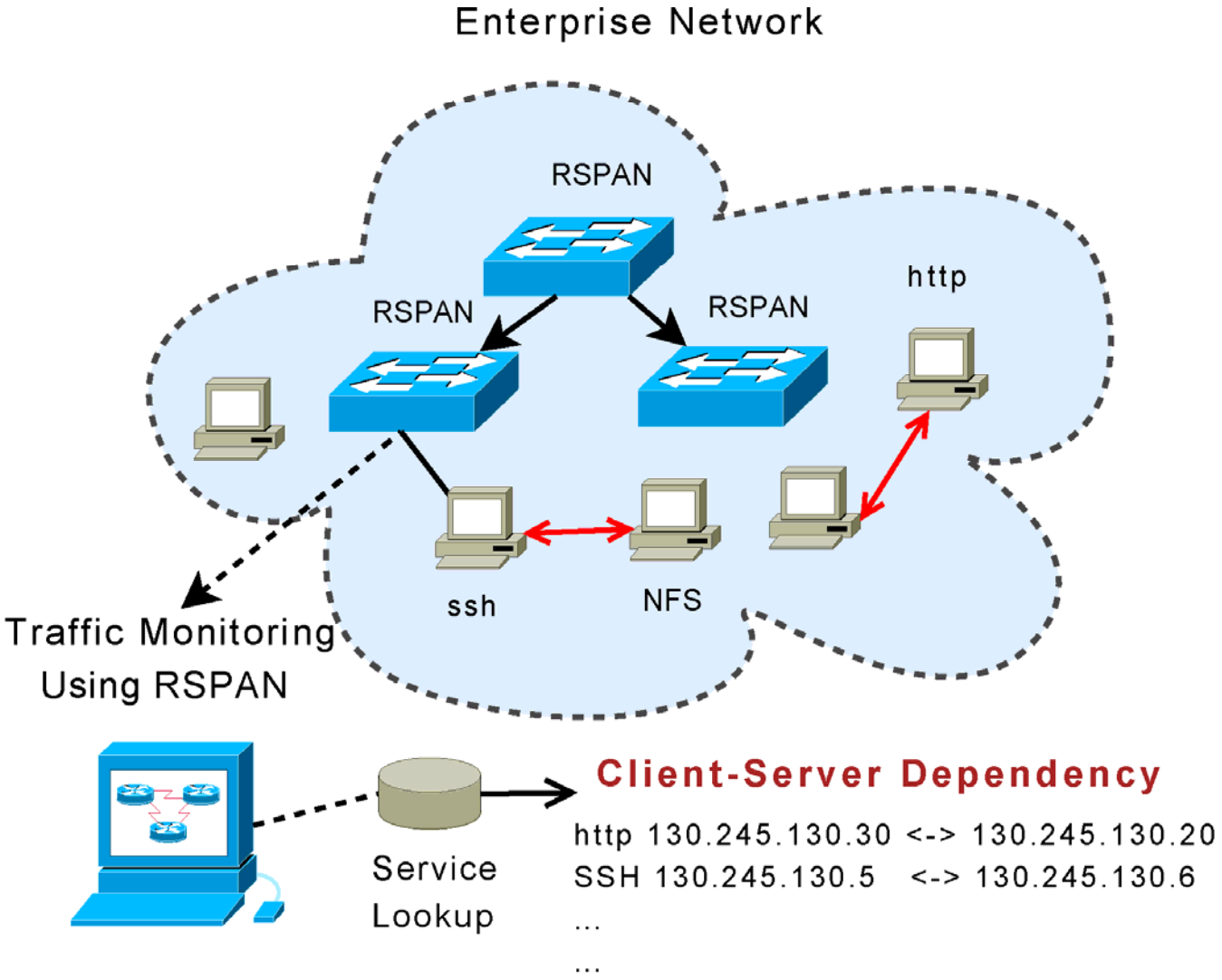
System Design: Function Blocks



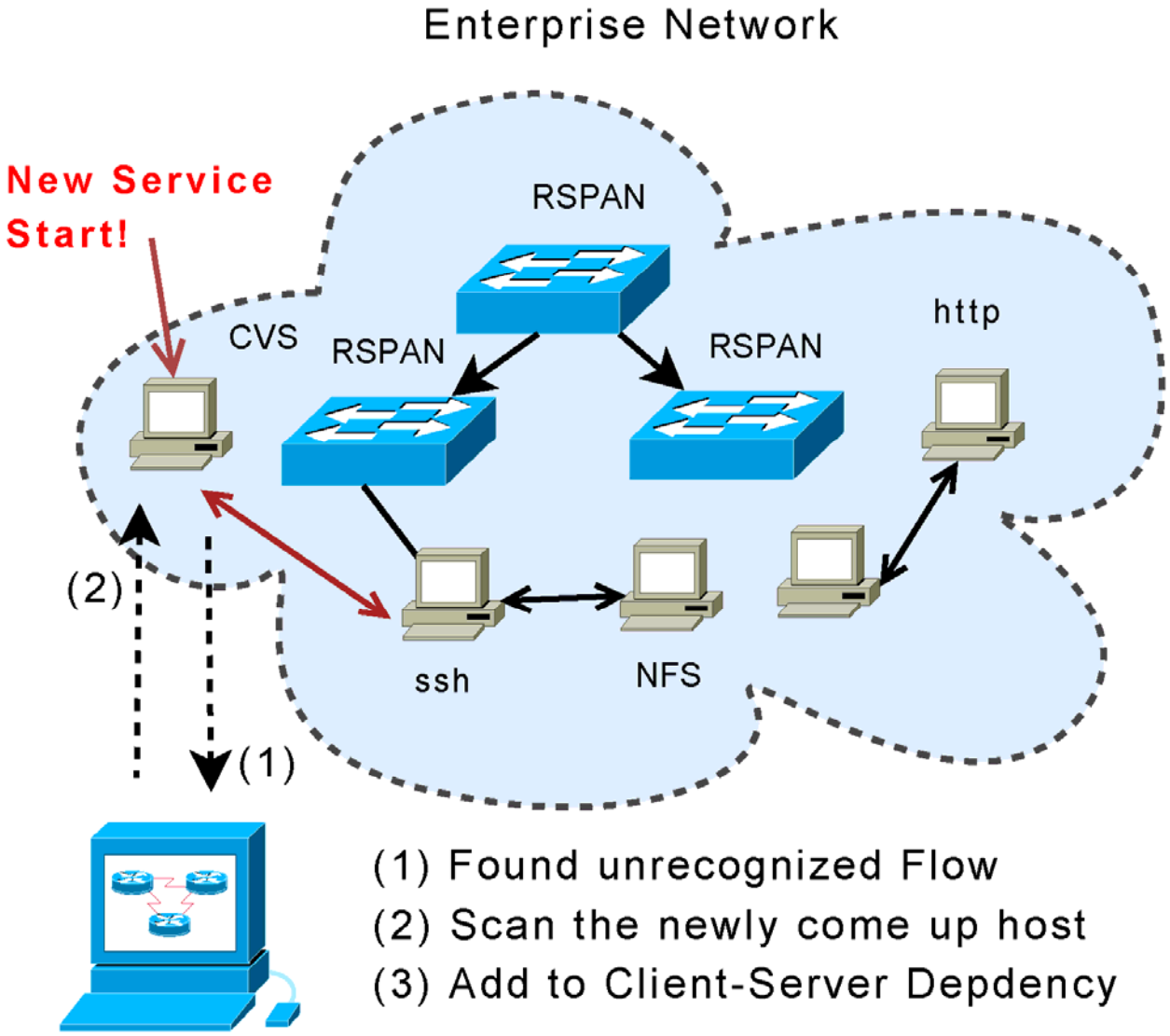
Step1: Nmap Initial Scan



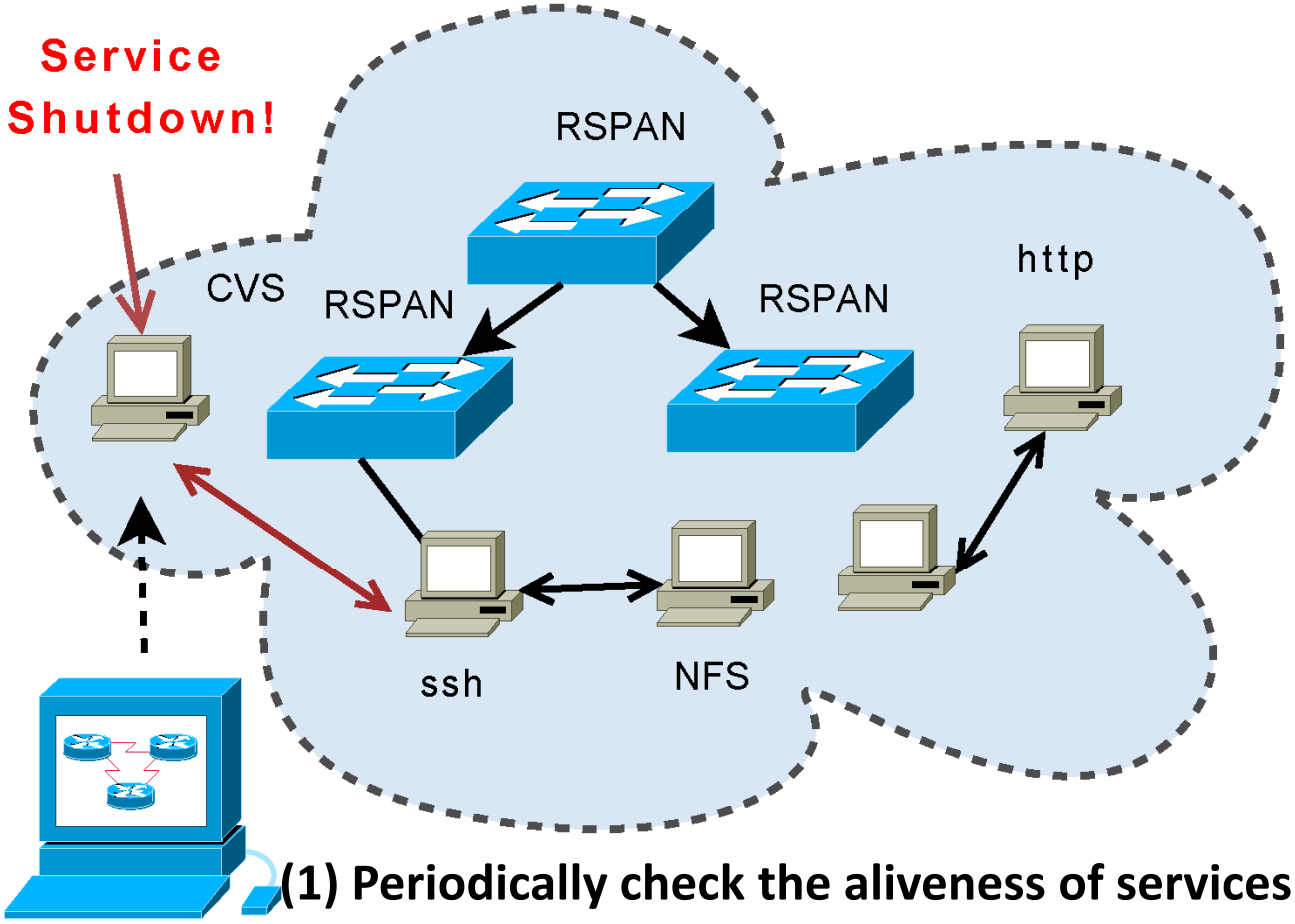
Step2: Traffic Monitoring



Step3: Detect New Services



Step4: Detect Obsolete Service



(1) Periodically check the aliveness of services

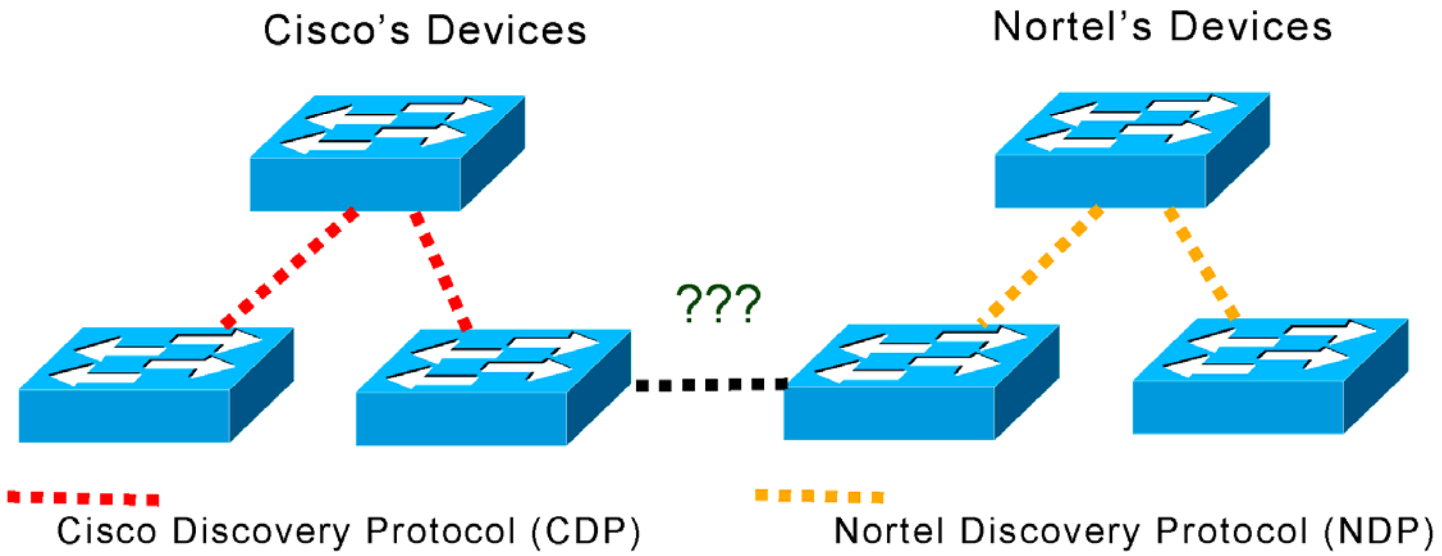
(2) If the service was detected as shutdown, remove it from our list

Talk outline

- Service Discovery
- System Design
- Network Topology Discovery
- Experiments
- Conclusion

L2 Network Topology Discovery

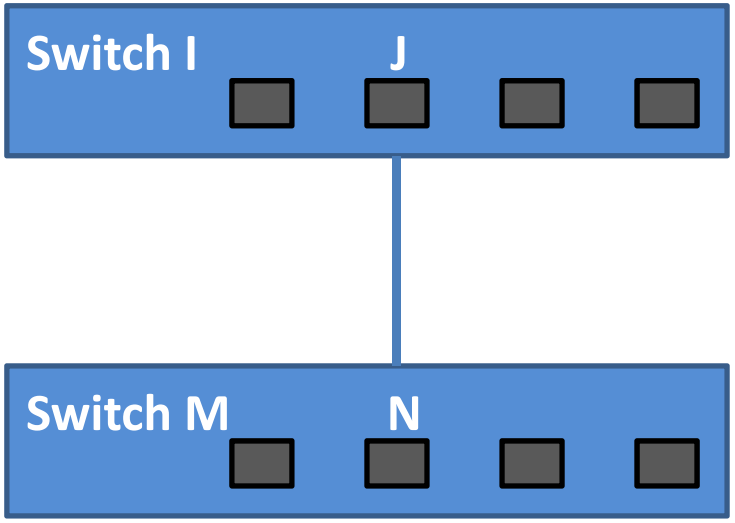
- Discovery Protocols (DP) are designed to share information between neighboring network devices, but…



Vendors ' proprietary discovery protocols can't communicate with others!

L2 Network Topology Discovery

- Solution: A vendor-independent Discovery Method



REACH[I, J] := the set of MAC addresses that switch I will forward using its port J.

1. Reachable through port J
If REACH[I,J] contains the MAC address of some ports in switch M

2. Directed Connected
Intersection(REACH[N,J], REACH[M,N]) = empty and condition 1

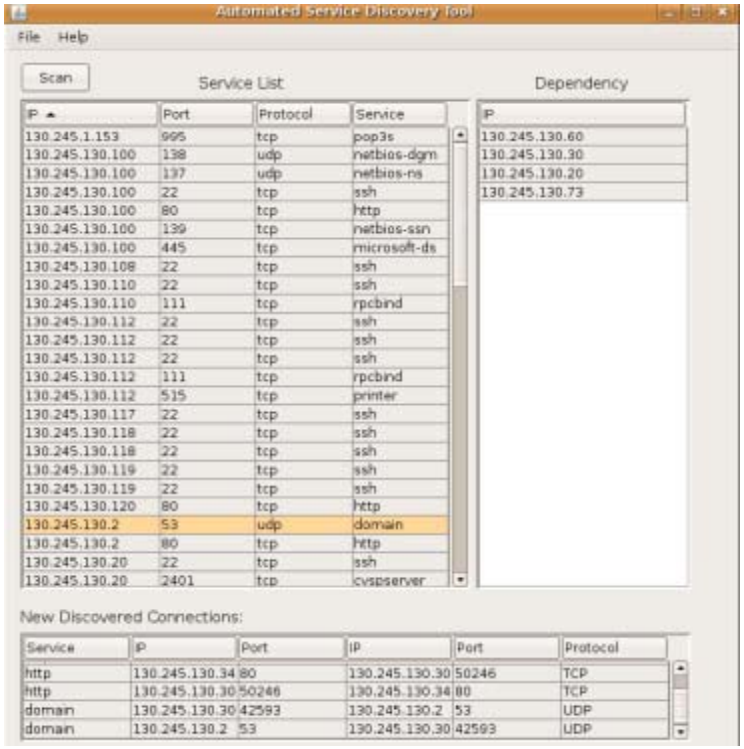
Talk outline

- Service Discovery
- System Design
- Network Topology Discovery
- Experiments
- Conclusion

Experiments

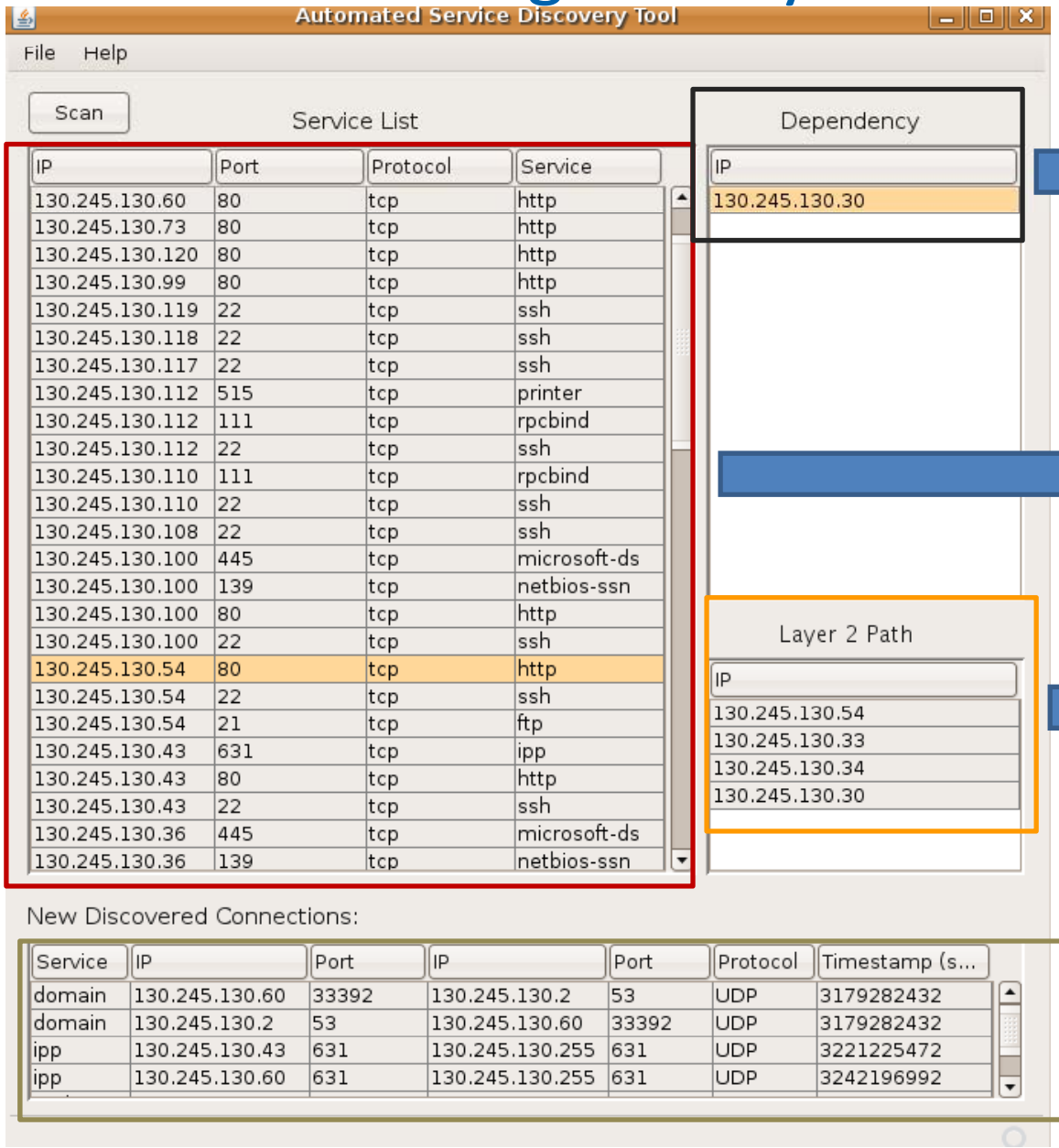
- The program was tested in ECSL lab
- The initial full Nmap scan time: 2 minutes.
- The rescan interval = 10 min to detect obsolete services.

IP range	# of PCs	# of Services
130.245.130.1 / 24	20	73



Program Screenshot

Program Layout



Host Dependency
Shows who connects to 130.245.30.2, port 53

Service Map

Layer 2 Path

New Discovered Connections

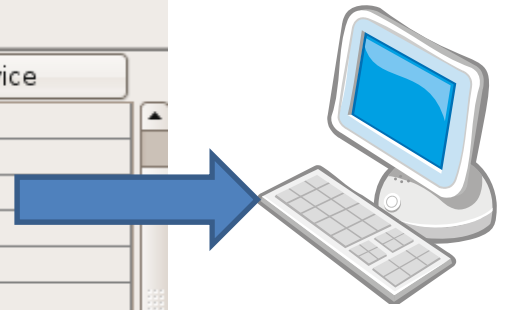
What services are active in the network?

Network Service Map

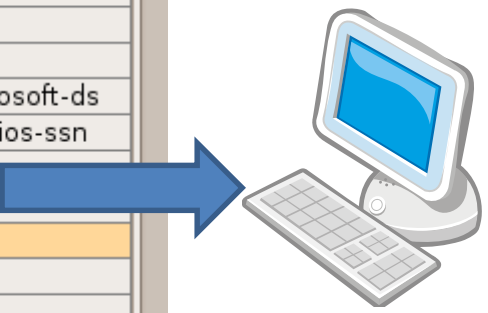
Scan

Service List

IP	Port	Protocol	Service
130.245.130.60	80	tcp	http
130.245.130.73	80	tcp	http
130.245.130.120	80	tcp	http
130.245.130.99	80	tcp	http
130.245.130.119	22	tcp	ssh
130.245.130.118	22	tcp	ssh
130.245.130.117	22	tcp	ssh
130.245.130.112	515	tcp	printer
130.245.130.112	111	tcp	rpcbind
130.245.130.112	22	tcp	ssh
130.245.130.110	111	tcp	rpcbind
130.245.130.110	22	tcp	ssh
130.245.130.108	22	tcp	ssh
130.245.130.100	445	tcp	microsoft-ds
130.245.130.100	139	tcp	netbios-ssn
130.245.130.100	80	tcp	http
130.245.130.100	22	tcp	ssh
130.245.130.54	80	tcp	http
130.245.130.54	22	tcp	ssh
130.245.130.54	21	tcp	ftp
130.245.130.43	631	tcp	ipp
130.245.130.43	80	tcp	http
130.245.130.43	22	tcp	ssh
130.245.130.36	445	tcp	microsoft-ds
130.245.130.36	139	tcp	netbios-ssn



Service	Port
http	80
ssh	22
ftp	21



Service	Port
https	443
ms-ds	445
rpcbind	111

Who connect to the service?

Network Service Map + Dependency

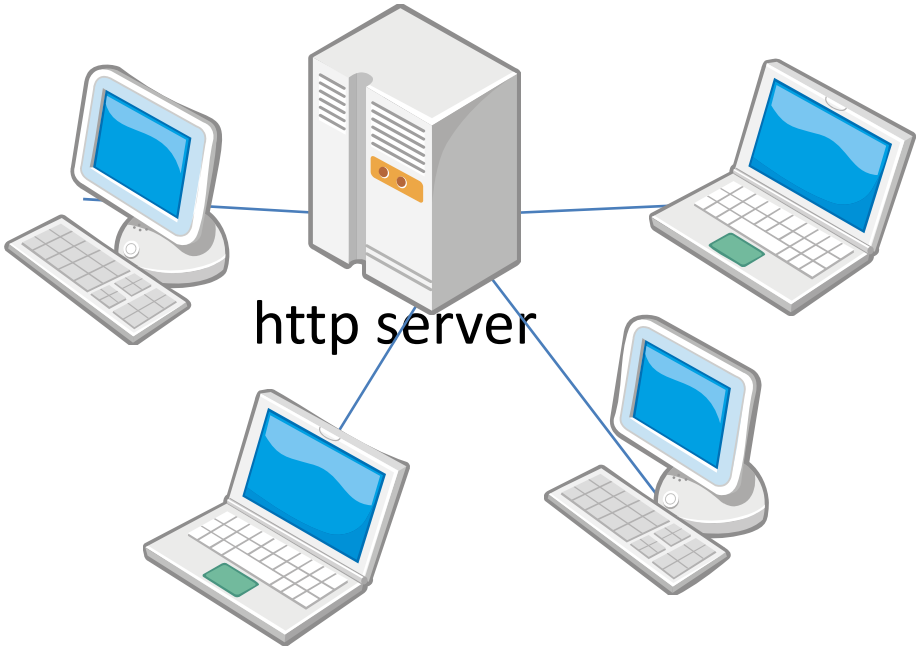
Scan

Service List

IP	Port	Protocol	Service
130.245.130.60	80	tcp	http
130.245.130.73	80	tcp	http
130.245.130.120	80	tcp	http
130.245.130.99	80	tcp	http
130.245.130.119	22	tcp	ssh
130.245.130.118	22	tcp	ssh
130.245.130.117	22	tcp	ssh
130.245.130.112	515	tcp	printer
130.245.130.112	111	tcp	rpcbind
130.245.130.112	22	tcp	ssh
130.245.130.110	111	tcp	rpcbind
130.245.130.110	22	tcp	ssh
130.245.130.108	22	tcp	ssh
130.245.130.100	445	tcp	microsoft-ds
130.245.130.100	139	tcp	netbios-ssn
130.245.130.100	80	tcp	http
130.245.130.100	22	tcp	ssh
130.245.130.54	80	tcp	http
130.245.130.54	22	tcp	ssh
130.245.130.54	21		
130.245.130.43	631		
130.245.130.43	80		
130.245.130.43	22		
130.245.130.36	445		
130.245.130.36	139		

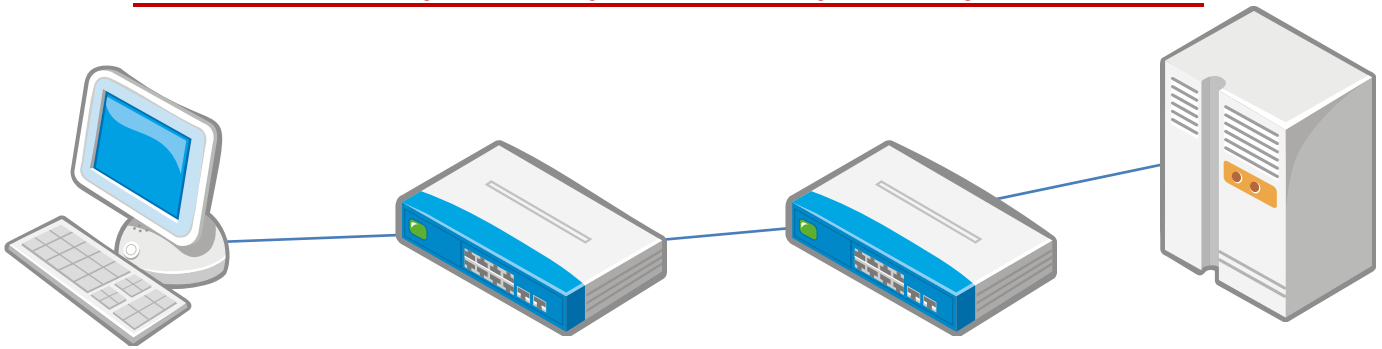
Dependency

IP
130.245.130.60
130.245.130.30
130.245.130.20
130.245.130.73



How do they connect to it?

Service Map + Dependency + Layer2 Path



Scan

Service List

IP	Port	Protocol	
130.245.130.60	80	tcp	
130.245.130.73	80	tcp	
130.245.130.120	80	tcp	
130.245.130.99	80	tcp	
130.245.130.119	22	tcp	
130.245.130.118	22	tcp	
130.245.130.117	22	tcp	
130.245.130.112	515	tcp	
130.245.130.112	111	tcp	
130.245.130.112	22	tcp	
130.245.130.110	111	tcp	
130.245.130.110	22	tcp	
130.245.130.108	22	tcp	ssh
130.245.130.100	445	tcp	microsoft-ds
130.245.130.100	139	tcp	netbios-ssn
130.245.130.100	80	tcp	http
130.245.130.100	22	tcp	ssh
130.245.130.54	80	tcp	http
130.245.130.54	22	tcp	ssh
130.245.130.54	21	tcp	ftp

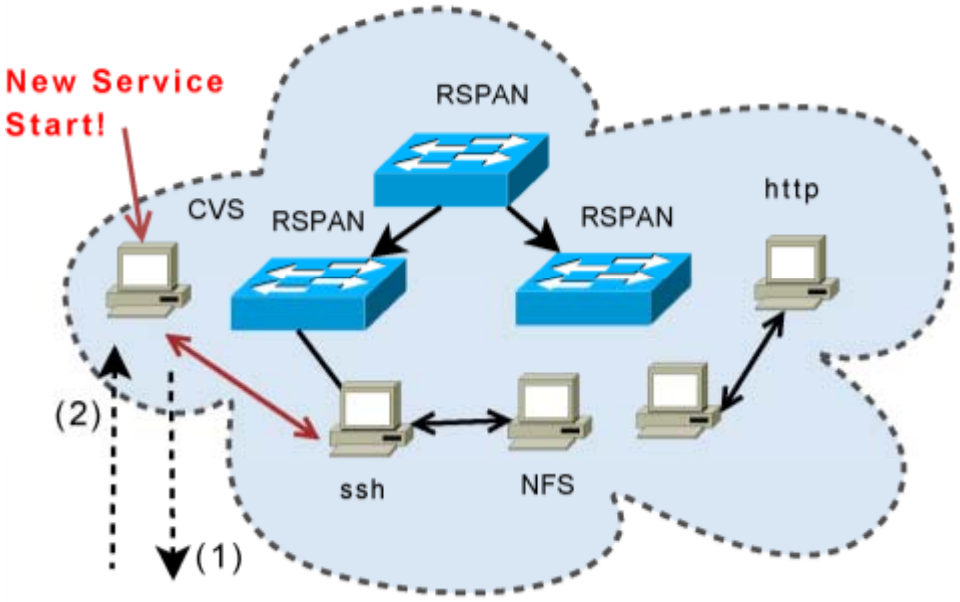
Dependency

IP
130.245.130.30

Layer 2 Path

IP
130.245.130.54
130.245.130.33
130.245.130.34
130.245.130.30

New Discovered Connections



Format:

Service	IP	Port	IP	Port	Protocol	Timestamp

New Discovered Connections:

Service	IP	Port	IP	Port	Protocol	Timestamp (s...
domain	130.245.130.60	33392	130.245.130.2	53	UDP	3179282432
domain	130.245.130.2	53	130.245.130.60	33392	UDP	3179282432
ipp	130.245.130.43	631	130.245.130.255	631	UDP	3221225472
ipp	130.245.130.60	631	130.245.130.255	631	UDP	3242196992

Conclusion and Future Work

- Implement L2 topology algorithm
- Improve current user interface
- Integrated the switch information and L2 topology into current user interface
- Discover L3 topology and integrate with L2

Thank You