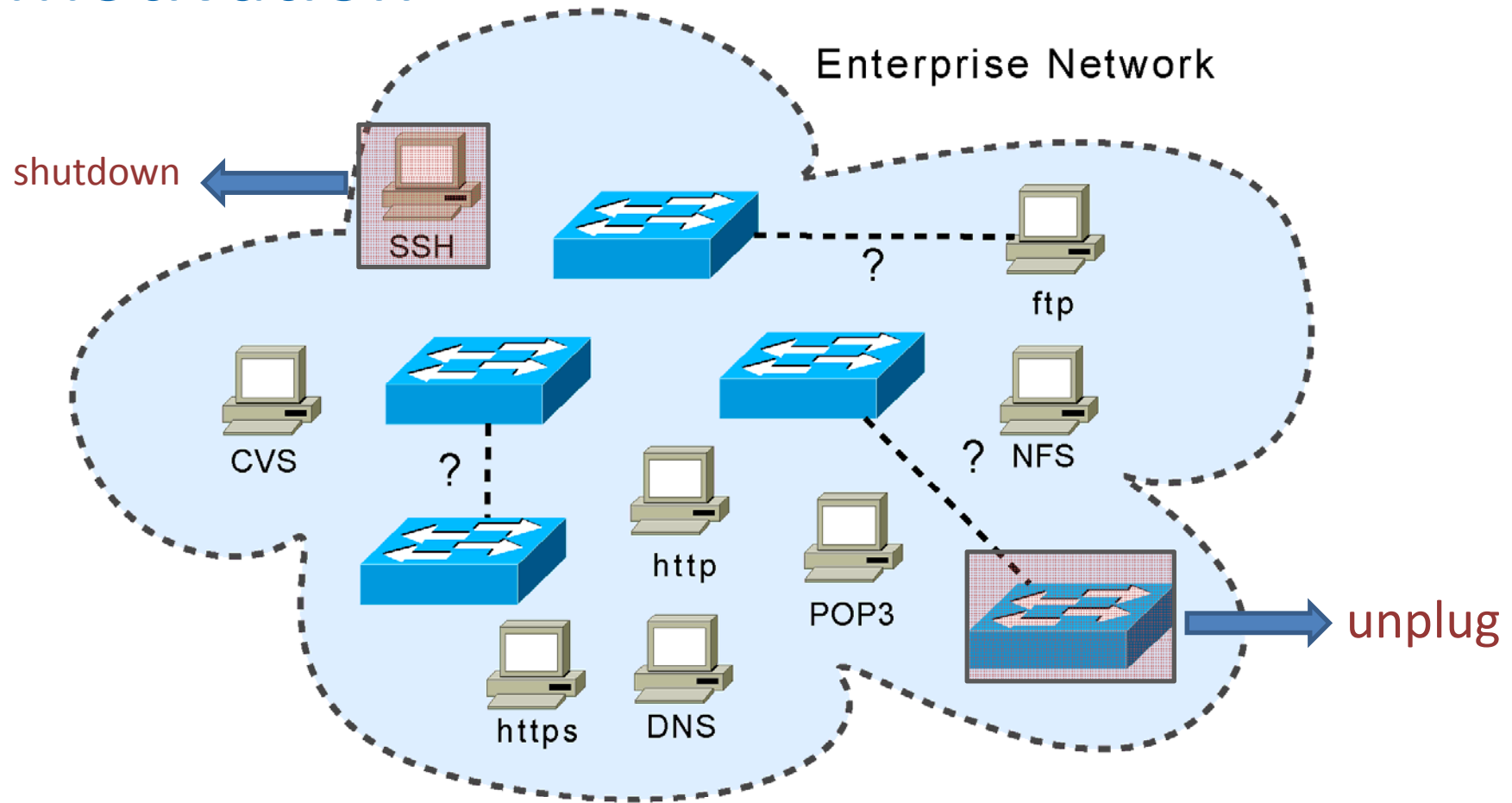


Automated Service Discovery for Enterprise Network Management

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



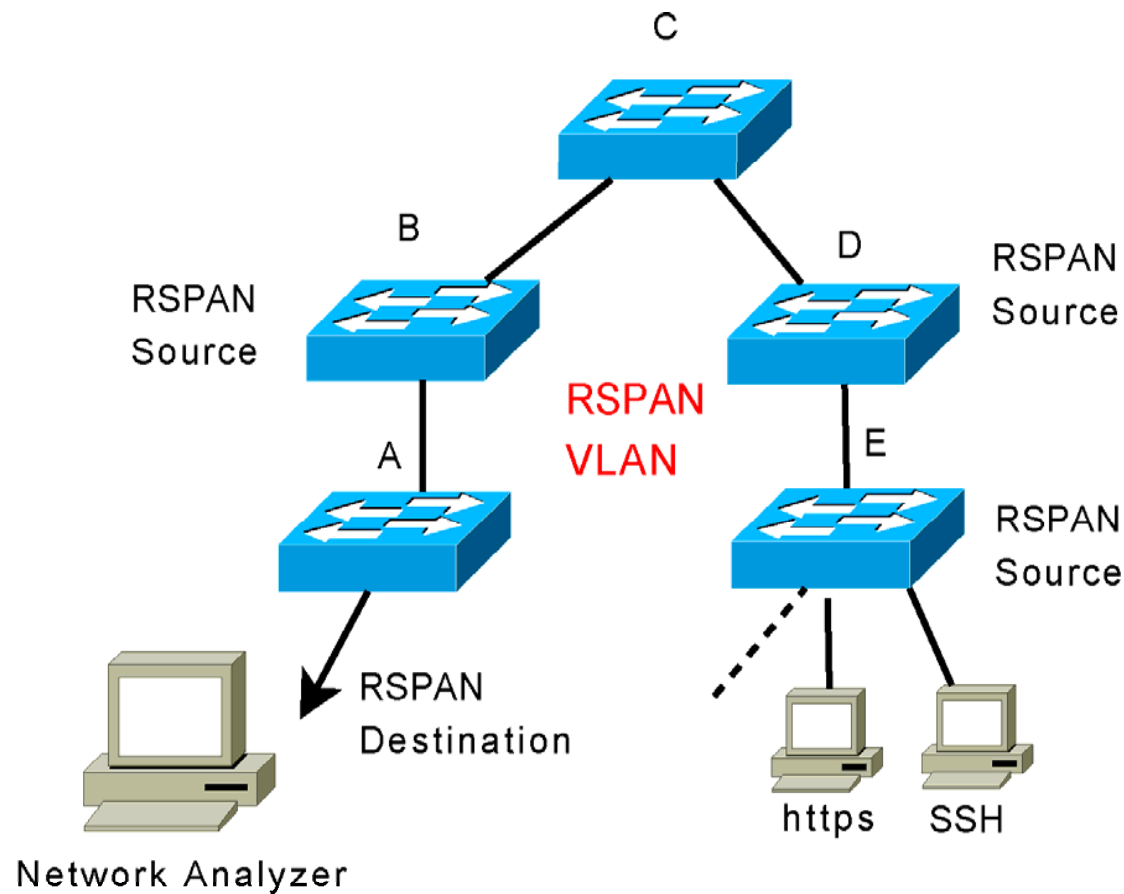
- Service Discovery
- System Design
- Network Topology Discovery
- Experiments
- Conclusion

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 RSPAN 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

No.	Time	Source	Destination	Protocol	Info
181	22.420567	130.245.132.177	130.245.134.28	TCP	35303 > ipp [ACK] Seq=473 Ack=...
182	22.450662	130.245.132.177	130.245.134.28	TCP	35302 > ipp [ACK] Seq=473 Ack=...
183	22.458657	130.245.132.177	130.245.134.28	TCP	35303 > ipp [ACK] Seq=473 Ack=...
184	24.203435	Cisco_22:1f:8a	Spanning-tree-(for	STP	Conf. Root = 32768/00:02:b9:2...
185	26.014463	130.245.132.170	130.245.132.255	CUPS	ipp://130.245.132.170:631/pr...
186	26.020480	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID...
187	26.050776	130.245.132.139	130.245.132.112	TCP	60384 > distinct [PSH, ACK] Seq=...
188	26.051191	130.245.132.112	130.245.132.139	TCP	distinct > 60384 [PSH, ACK] Seq=...
189	26.051214	130.245.132.139	130.245.132.112	TCP	60384 > distinct [ACK] Seq=30...
190	26.206265	Cisco_22:1f:8a	Spanning-tree-(for	STP	Conf. Root = 32768/00:02:b9:2...
191	26.226433	130.245.132.139	130.245.132.112	TCP	60384 > distinct [PSH, ACK] Seq=...
192	26.226832	130.245.132.112	130.245.132.139	TCP	distinct > 60384 [PSH, ACK] Seq=...
193	26.226833	130.245.132.139	130.245.132.112	TCP	60384 > distinct [ACK] Seq=3...
194	26.376985	130.245.132.139	130.245.132.112	TCP	60384 > distinct [PSH, ACK] Seq=...
195	26.377386	130.245.132.112	130.245.132.139	TCP	distinct > 60384 [PSH, ACK] Seq=...
196	26.377409	130.245.132.139	130.245.132.112	TCP	60384 > distinct [ACK] Seq=40...
197	26.562541	130.245.132.139	130.245.132.112	TCP	60384 > distinct [PSH, ACK] Seq=...
198	26.562934	130.245.132.112	130.245.132.139	TCP	distinct > 60384 [PSH, ACK] Seq=...
199	26.562957	130.245.132.139	130.245.132.112	TCP	60384 > distinct [ACK] Seq=45...
200	28.209029	Cisco_22:1f:8a	Spanning-tree-(for	STP	Conf. Root = 32768/00:02:b9:2...

Frame 192 (279 bytes on wire, 279 bytes captured)

- Ethernet II, Src: Dell_b1:43:82 (00:14:22:b1:43:82), Dst: Dell_cf:01:3f (00:12:3f:cf:01:3f)
- Internet Protocol, Src: 130.245.132.112 (130.245.132.112), Dst: 130.245.132.139 (130.245.132.139)
- Transmission Control Protocol, Src Port: distinct (9999), Dst Port: 60384 (60384), Seq: 128...
- Data (213 bytes)

0020 84 8b 27 0f eb e0 07 2e 28 3f d7 be 31 60 80 18 ..'..... (?..1..

0030 00 do b2 aa 00 00 01 01 08 0a 1e 71 4b 52 04 11qnr..

0040 b7 24 48 54 54 50 2f 31 2e 31 20 33 30 34 20 4e .\$.HTTP/1.1 304 N

0050 6f 74 20 4d 6f 64 69 66 69 65 64 0d 0a 44 61 71 et Modif ied..Dat

Transmission Control Protocol (tcp),... Packets: 200 Displayed: 200 Marke... Profile: Default

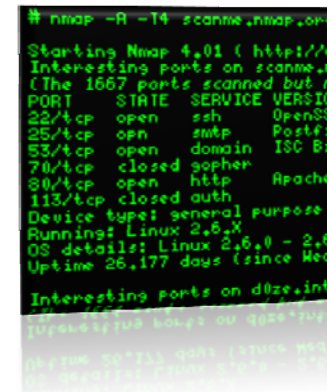
http connection
running on port 9999

Wireshark can not detect
it's a http protocol!

Packet content

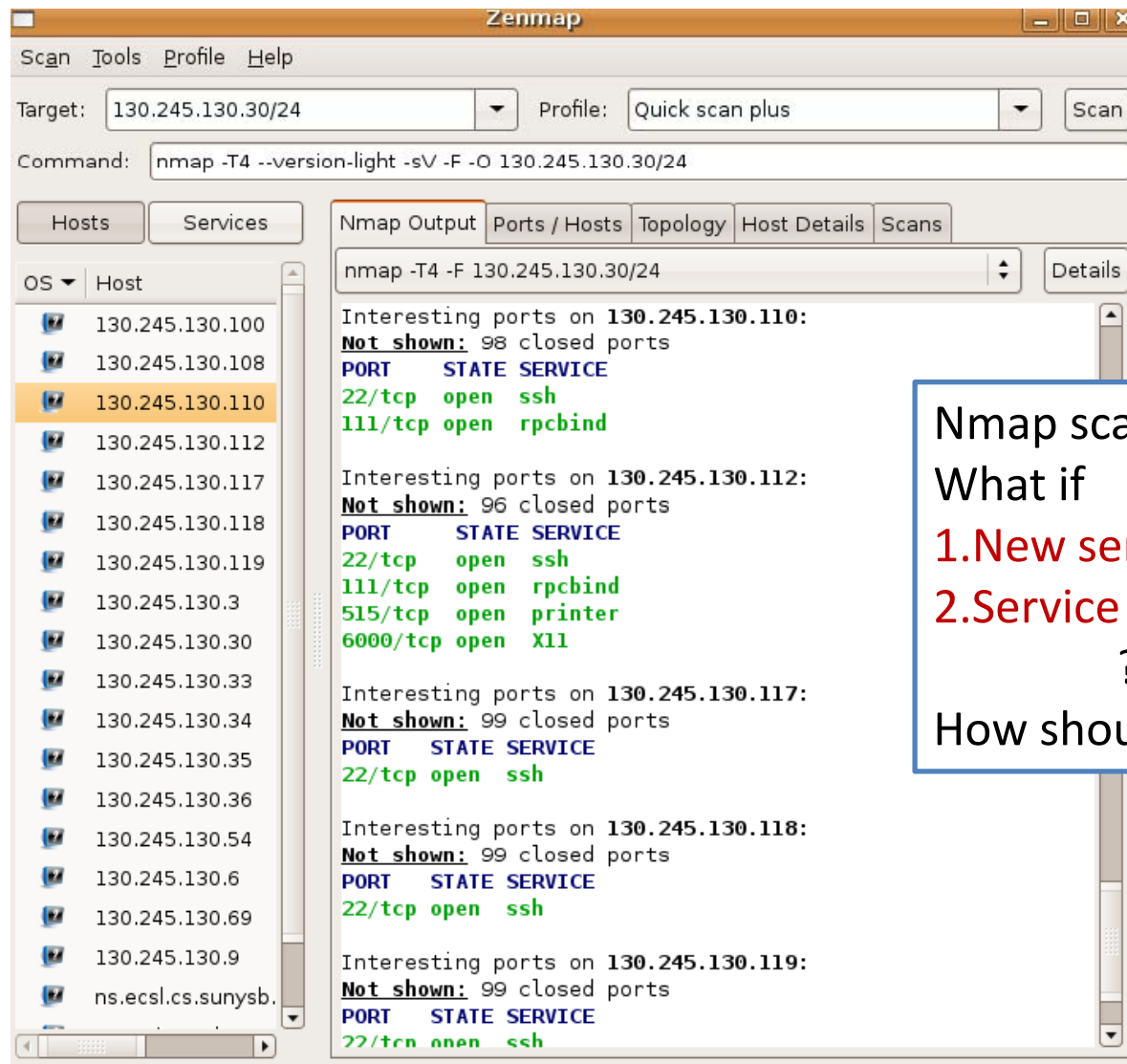
Service Discovery: Nmap

- 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 -R -T4 scanme.nmap.org
Starting Nmap 4.01 ( http://w
Interesting ports on scanme.o
(The 1667 ports scanned but o
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH
25/tcp    open  smtp     Postfix
53/tcp    open  domain   ISC BIND
70/tcp    closed sopher
80/tcp    open  http     Apache
113/tcp   closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.0 - 2.6
Uptime 26,177 days (since Wed
Interesting ports on d0ze.inte
[...]
```

Nmap GUI: Zenmap



Nmap scan result...
What if
1. New service coming up
2. Service was shutdown
??
How should we scan?

Nmap GUI: Zenmap

Scan Tools Profile Help

Target: 130.245.130.30/24 Profile: Quick scan plus Scan

Command: nmap -T4 --version-light -sV -F -O 130.245.130.30/24

Hosts Services Nmap Output Ports / Hosts **Topology** Host Details Scans

Hosts Viewer Fisheye Controls

OS Host

130.245.130.35 130.245.130.34 130.245.130.33 130.245.130.119 130.245.130.54 130.245.130.112 130.245.130.69 130.245.130.118 130.245.130.9 130.245.130.108 130.245.130.117 130.245.130.3 130.245.130.30 130.245.130.6 130.245.130.30 ns.ecsl.cs.sunysb.edu sequoia.ecsl.cs.sunysb.edu www.ecsl.cs.sunysb.edu localhost

Fisheye on ring 1.00 with interest factor 3.14 and spread factor 0.68

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

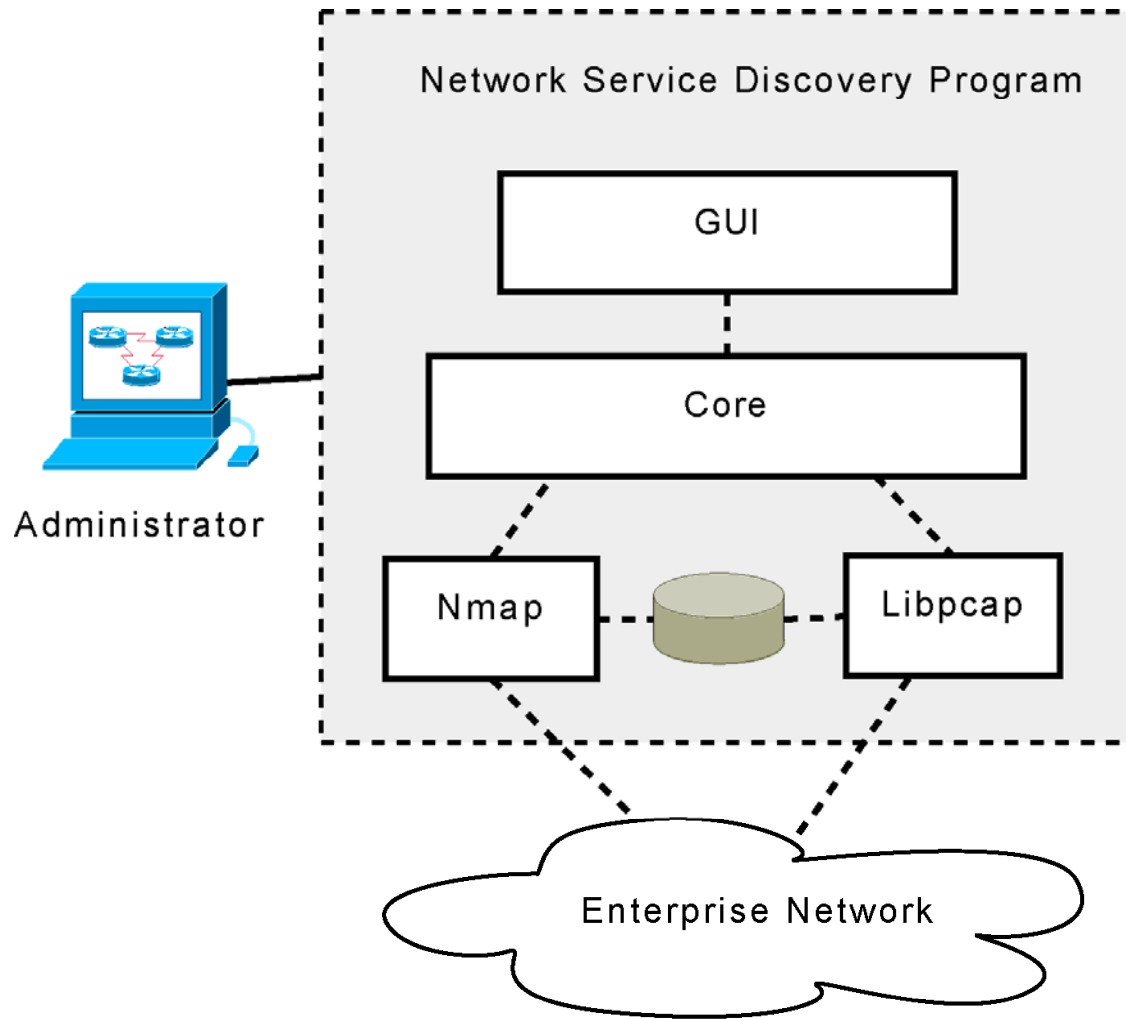
A tool that can

- 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

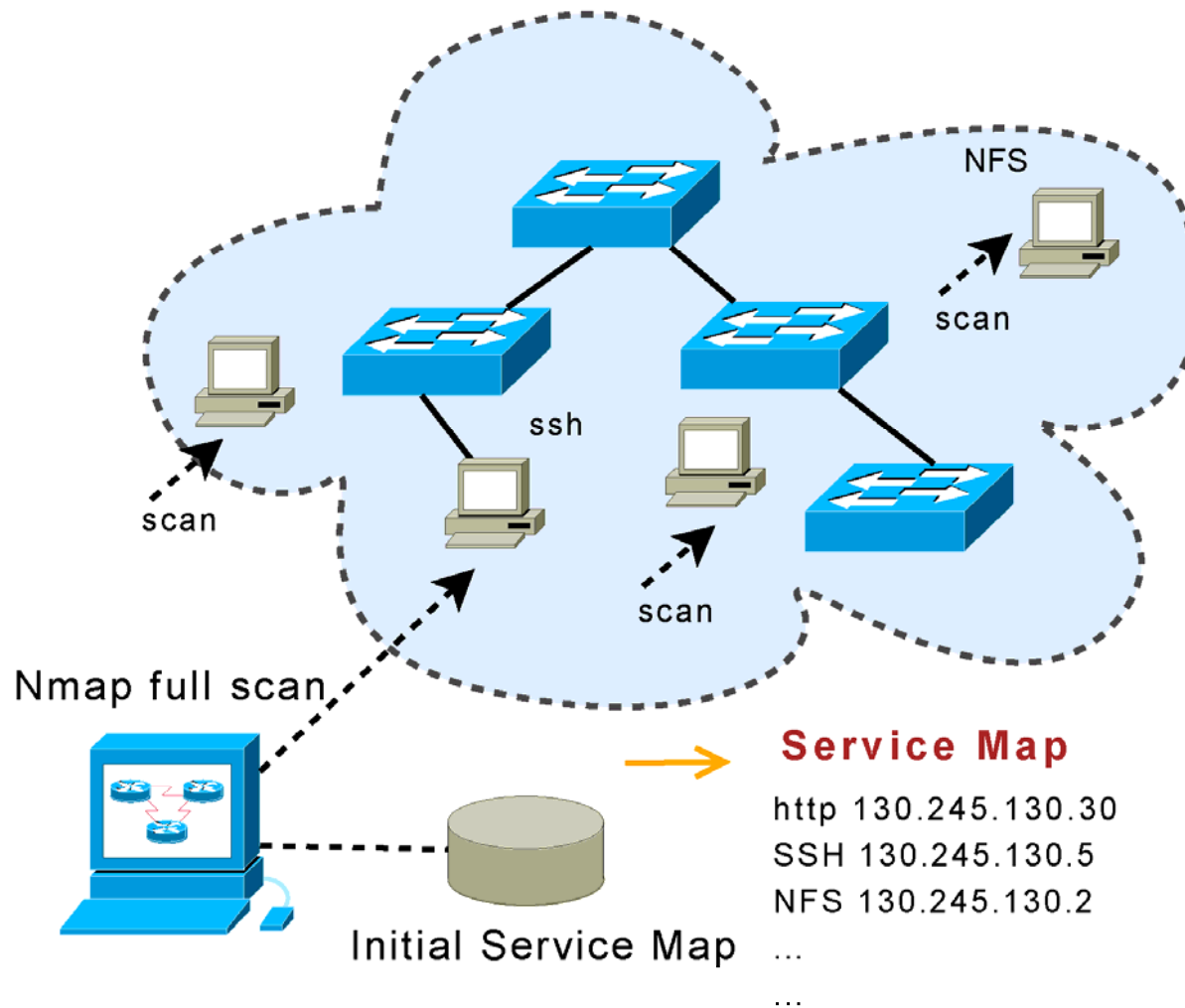
- Service Discovery
-  ■ System Design
- Network Topology Discovery
- Experiments
- Conclusion

System Design: Function Blocks



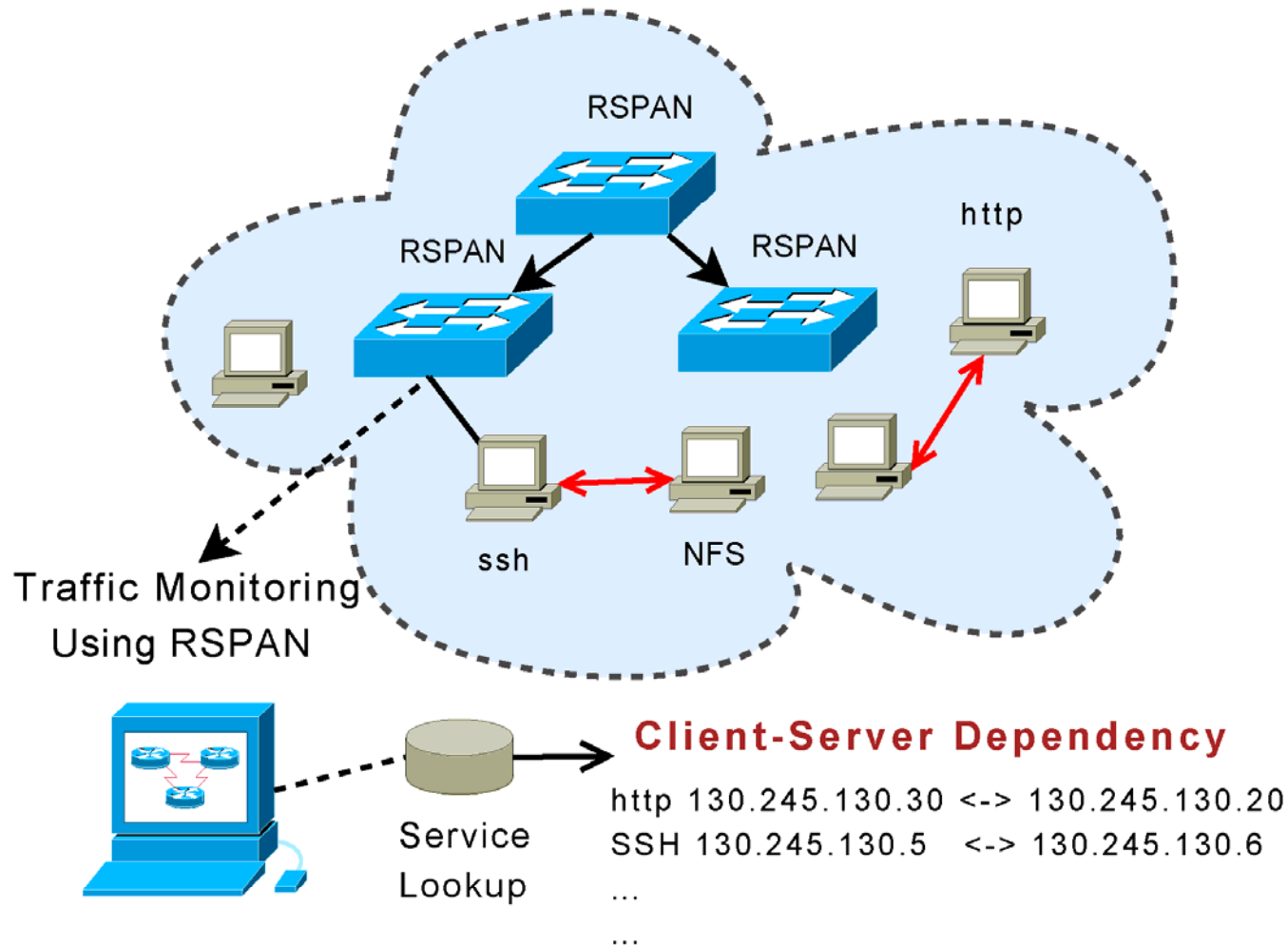
Step1: Nmap Initial Scan

Enterprise Network



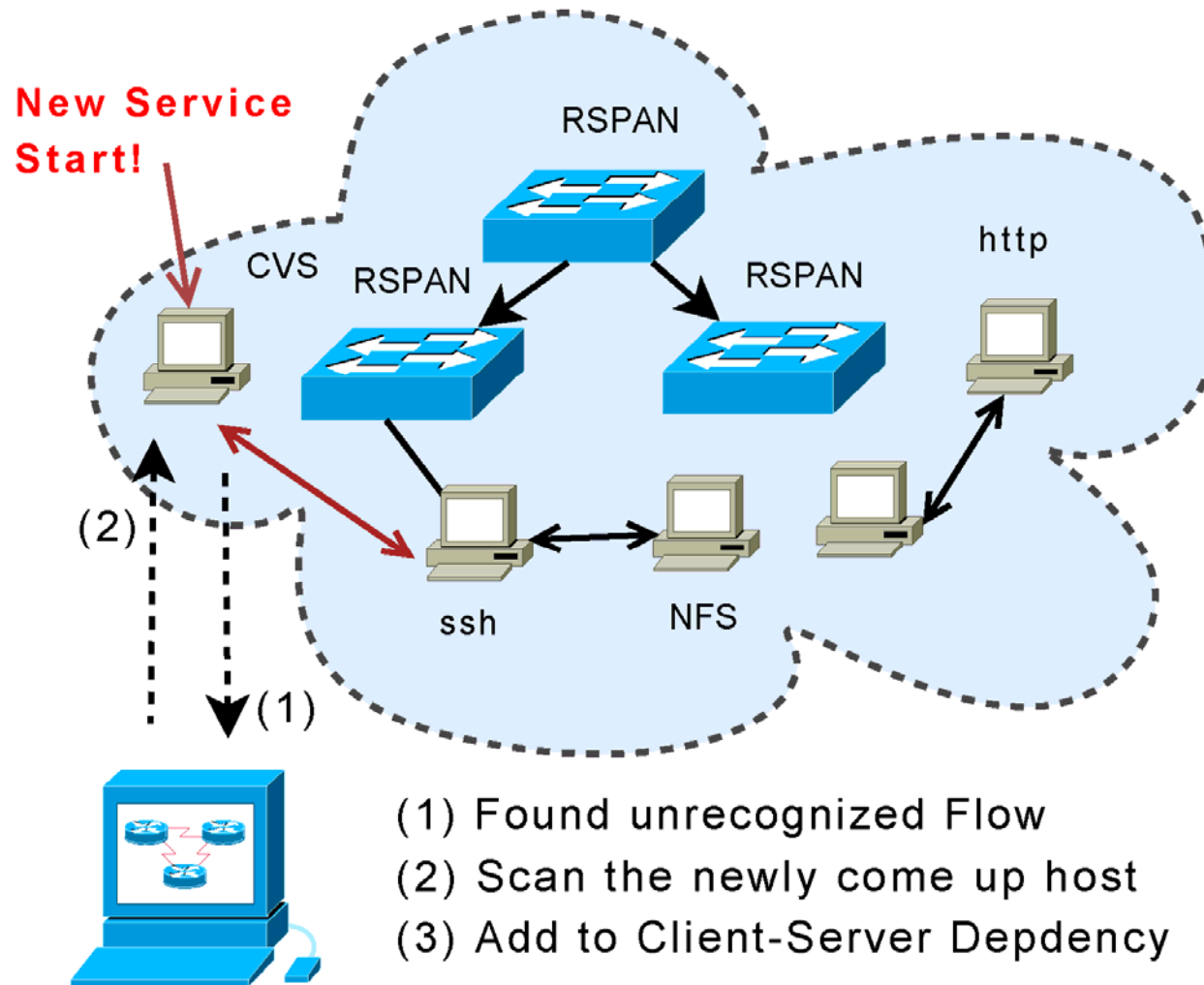
Step2: Traffic Monitoring

Enterprise Network



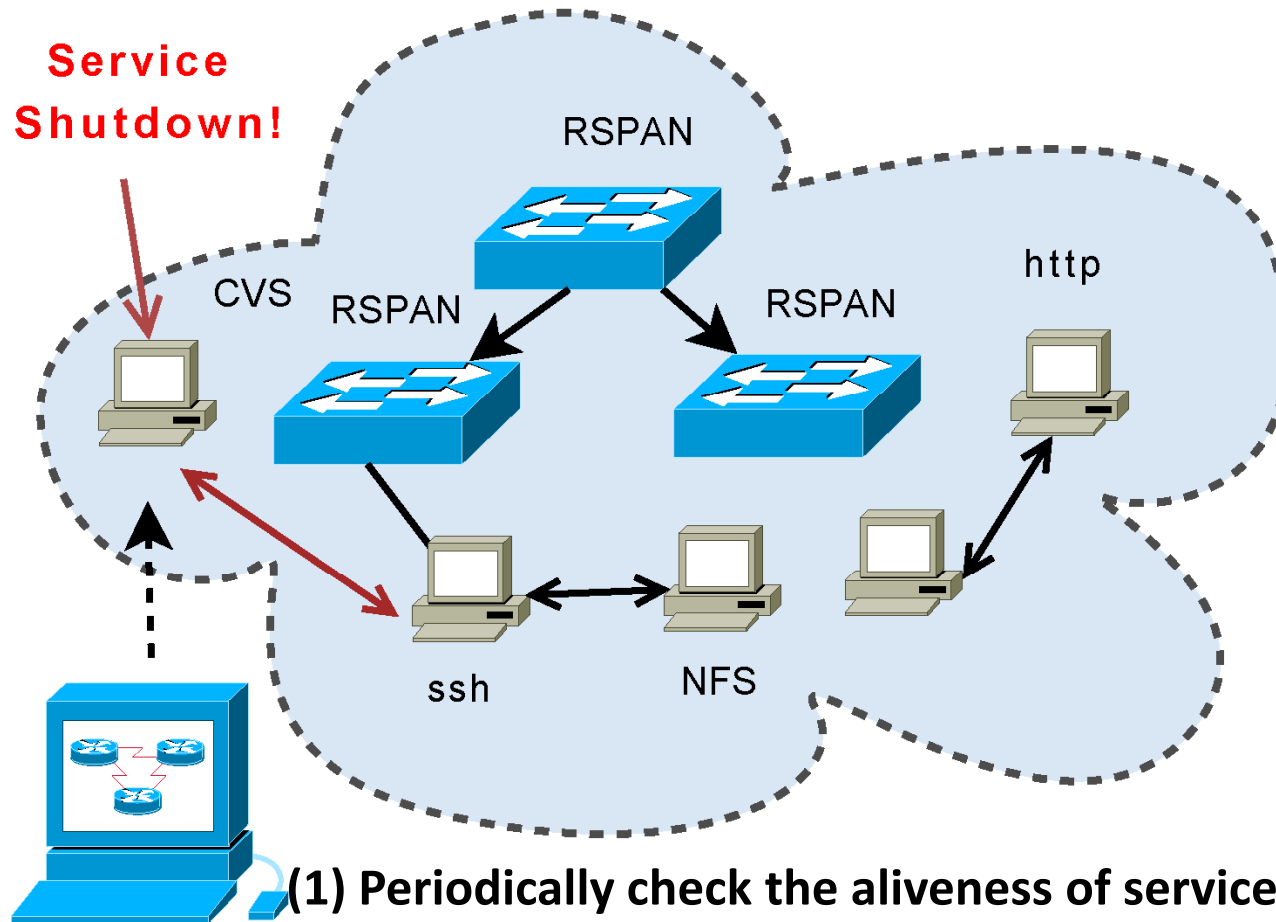
Step3: Detect New Services

Enterprise Network




Step4: Detect Obsolete Service

Enterprise Network



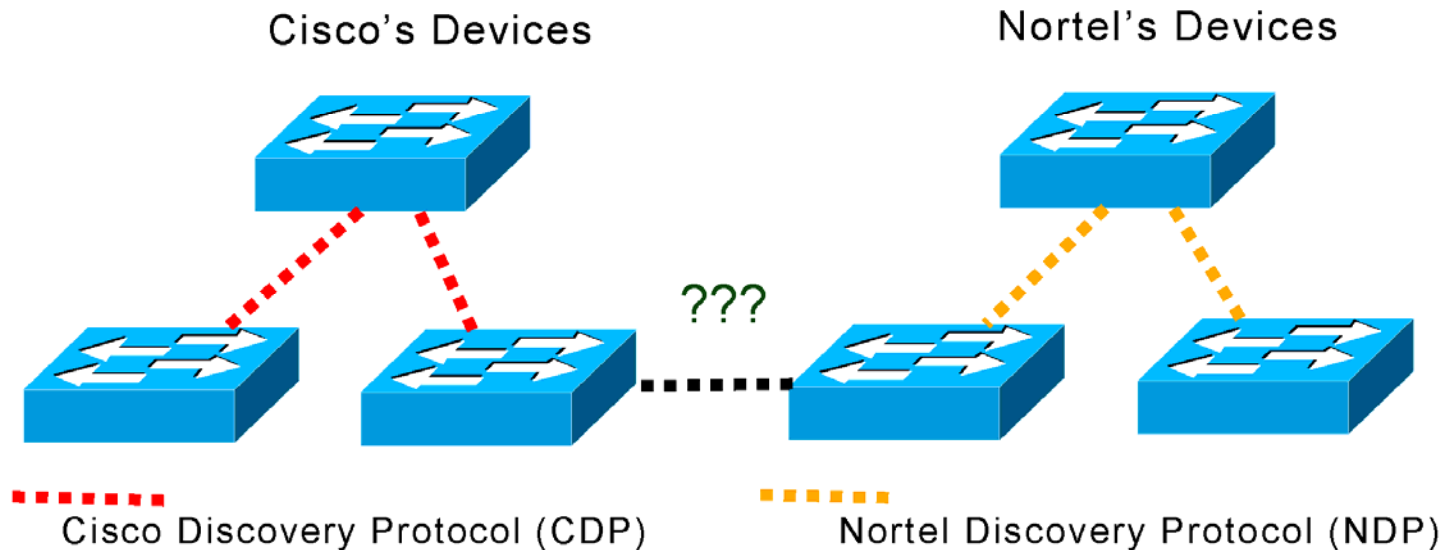
- (1) Periodically check the aliveness of services
- (2) If the service was detected as shutdown, remove it from our list

Talk outline

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

L2 Network Topology Discovery

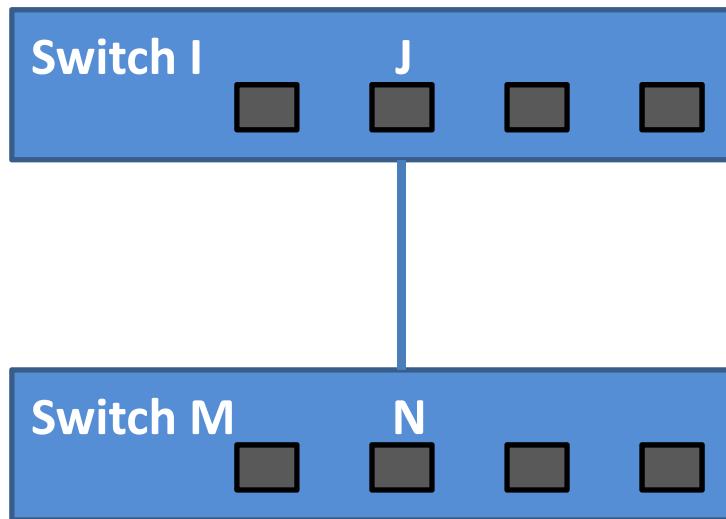
- Discovery Protocols (DP) are designed to share information between neighboring network devices, but...



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
 $Intersection(REACH[N, J], REACH[M, N]) = \text{empty}$ and condition 1

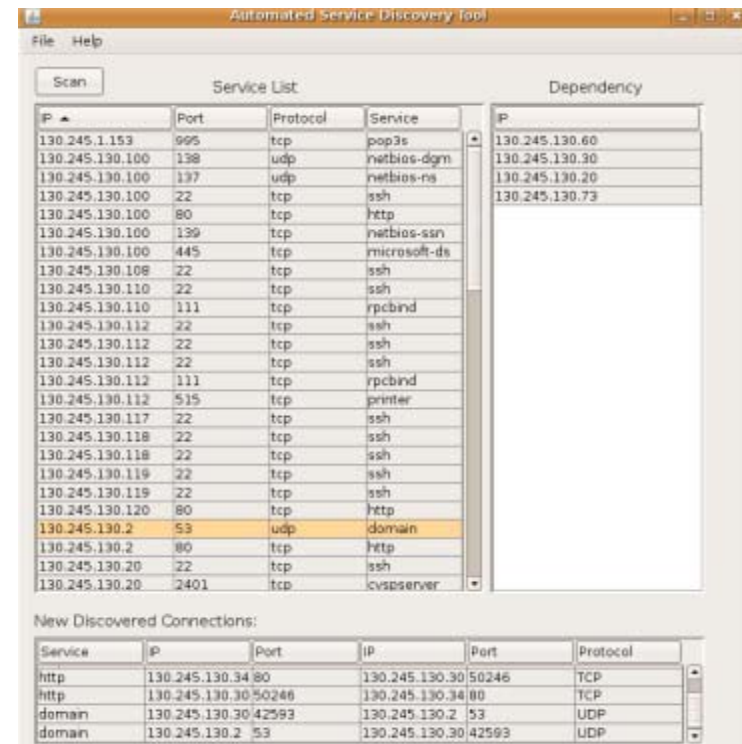
Talk outline

- 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

The screenshot shows the 'Automated Service Discovery Tool' interface. It features a 'Service List' table with columns for IP, Port, Protocol, and Service. A red box highlights the entire table. To the right, there are three panels: 'Dependency' (with a highlighted IP of 130.245.130.30), 'Layer 2 Path' (with a list of IP addresses), and 'New Discovered Connections' (a table of connection details). Blue arrows point from text labels to these panels.

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	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

Host Dependency
Shows who connects to 130.245.30.2, port 53

Service Map

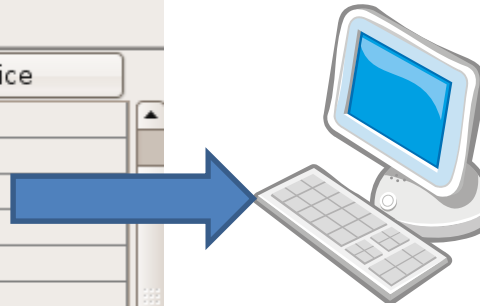
Layer 2 Path

New Discovered Connections

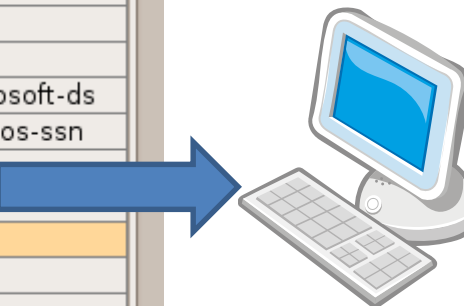
What services are active in the network?

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



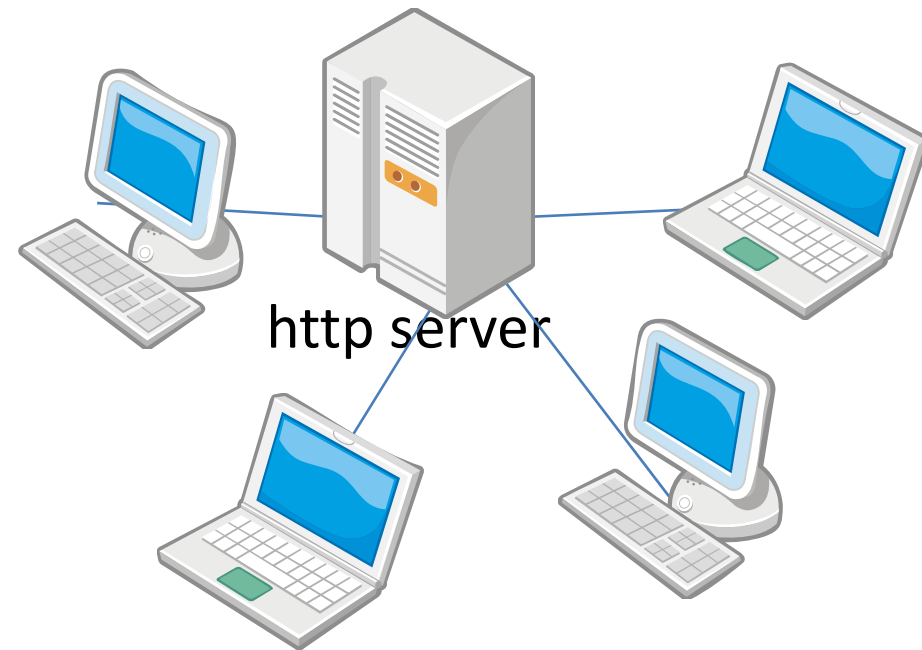
Service	Port
https	443
ms-ds	445
rpcbind	111

Who connect to the service?

Network Service Map + Dependency

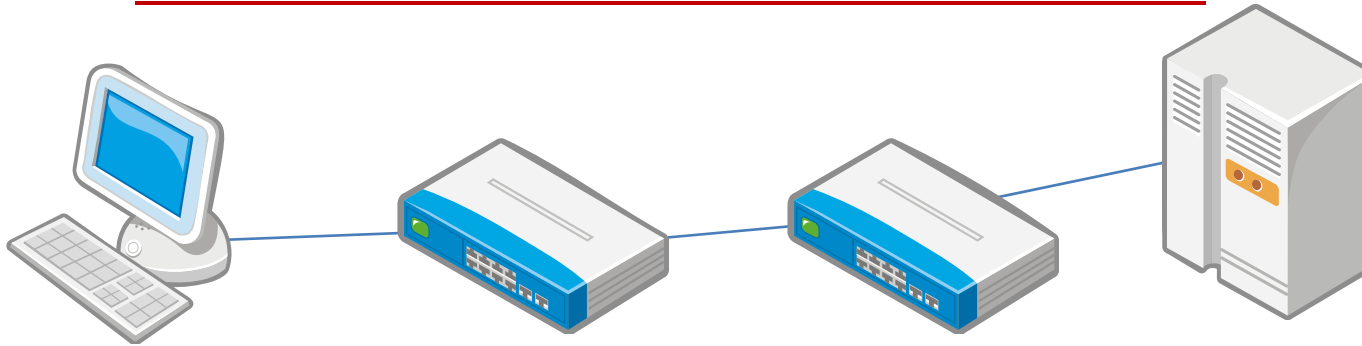
Scan			
Service List			
IP	Port	Protocol	Service
130.245.130.60	80	tcp	http
130.245.130.73	80	tcp	http
130.245.130.120	80	tcp	http
130.245.130.99	80	tcp	http
130.245.130.119	22	tcp	ssh
130.245.130.118	22	tcp	ssh
130.245.130.117	22	tcp	ssh
130.245.130.112	515	tcp	printer
130.245.130.112	111	tcp	rpcbind
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

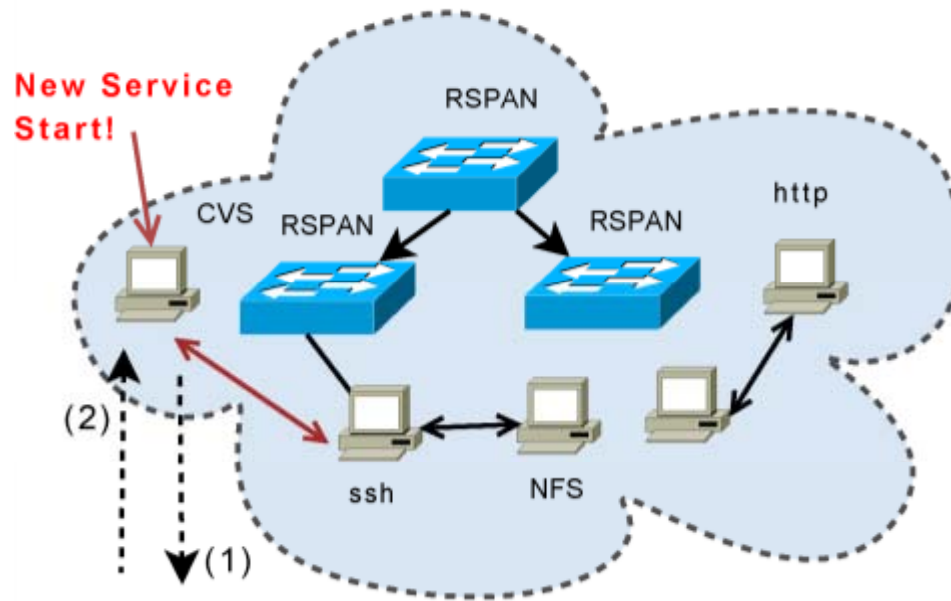


Service List		
IP	Port	Protocol
130.245.130.60	80	tcp
130.245.130.73	80	tcp
130.245.130.120	80	tcp
130.245.130.99	80	tcp
130.245.130.119	22	tcp
130.245.130.118	22	tcp
130.245.130.117	22	tcp
130.245.130.112	515	tcp
130.245.130.112	111	tcp
130.245.130.112	22	tcp
130.245.130.110	111	tcp
130.245.130.110	22	tcp
130.245.130.108	22	tcp
130.245.130.100	445	tcp
130.245.130.100	139	tcp
130.245.130.100	80	tcp
130.245.130.100	22	tcp
130.245.130.54	80	tcp
130.245.130.54	22	tcp
130.245.130.54	21	tcp

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